UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

# CCITT

COMITÉ CONSULTIVO
INTERNACIONAL
TELEGRÁFICO Y TELEFÓNICO

**LIBRO AZUL**

---

**TOMO VIII – FASCÍCULO VIII.8**

# REDES DE COMUNICACIÓN DE DATOS
# LA GUÍA

**RECOMENDACIONES X.500 A X.521**

---

**IX ASAMBLEA PLENARIA**
MELBOURNE, 14-25 DE NOVIEMBRE DE 1988

Ginebra 1989

UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

# CCITT

COMITÉ CONSULTIVO
INTERNACIONAL
TELEGRÁFICO Y TELEFÓNICO

**LIBRO AZUL**

---

**TOMO VIII – FASCÍCULO VIII.8**

# REDES DE COMUNICACIÓN DE DATOS
# LA GUÍA

**RECOMENDACIONES X.500 A X.521**

---

**IX ASAMBLEA PLENARIA**
MELBOURNE, 14-25 DE NOVIEMBRE DE 1988

Ginebra 1989

ISBN 92-61-03733-X

Impreso en Suiza

## CONTENIDO DEL LIBRO DEL CCITT EN VIGOR DESPUÉS DE LA NOVENA ASAMBLEA PLENARIA (1988)

## LIBRO AZUL

FASCÍCULO III.6 – Transmisión en línea de señales no telefónicas. Transmisión de señales radiofónicas y de televisión. Recomendaciones de las series H y J (Comisión de Estudio XV).

FASCÍCULO III.7 – Red digital de servicios integrados (RDSI). Estructura general y capacidades de servicio. Recomendaciones I.110 a I.257 (Comisión de Estudio XVIII).

FASCÍCULO III.8 – Red digital de servicios integrados (RDSI). Aspectos y funciones globales de la red, interfaces usuario-red de la RDSI. Recomendaciones I.310 a I.470 (Comisión de Estudio XVIII).

FASCÍCULO III.9 – Red digital de servicios integrados (RDSI). Interfaces entre redes y principios de mantenimiento. Recomendaciones I.500 a I.605 (Comisión de Estudio XVIII).

**Tomo IV**

FASCÍCULO IV.1 – Principios generales de mantenimiento: mantenimiento de los sistemas de transmisión y de los circuitos telefónicos internacionales. Recomendaciones M.10 a M.782 (Comisión de Estudio IV).

FASCÍCULO IV.2 – Mantenimiento de circuitos internacionales de telegrafía y de telefotografía y de circuitos internacionales arrendados. Mantenimiento de la red telefónica pública internacional. Mantenimiento de sistemas marítimos por satélite y de transmisión de datos. Recomendaciones M.800 a M.1375 (Comisión de Estudio IV).

FASCÍCULO IV.3 – Mantenimiento de circuitos internacionales para transmisiones radiofónicas y de televisión. Recomendaciones de la serie N (Comisión de Estudio IV).

FASCÍCULO IV.4 – Especificaciones de los aparatos de medida. Recomendaciones de la serie O (Comisión de Estudio IV).

**Tomo V** – Calidad de transmisión telefónica. Recomendaciones de la serie P (Comisión de Estudio XII).

**Tomo VI**

FASCÍCULO VI.1 – Recomendaciones generales sobre la conmutación y la señalización telefónicas. Funciones y flujos de información para los servicios de la RDSI. Suplementos. Recomendaciones Q.1 a Q.118 *bis* (Comisión de Estudio XI).

FASCÍCULO VI.2 – Especificaciones de los sistemas de señalización N.$^{os}$ 4 y 5. Recomendaciones Q.120 a Q.180 (Comisión de Estudio XI).

FASCÍCULO VI.3 – Especificaciones del sistema de señalización N.° 6. Recomendaciones Q.251 a Q.300 (Comisión de Estudio XI).

FASCÍCULO VI.4 – Especificaciones de los sistemas de señalización R1 y R2. Recomendaciones Q.310 a Q.490 (Comisión de Estudio XI).

FASCÍCULO VI.5 – Centrales digitales locales, de tránsito, combinadas e internacionales en redes digitales integradas y en redes mixtas analógico-digitales. Suplementos. Recomendaciones Q.500 a Q.554 (Comisión de Estudio XI).

FASCÍCULO VI.6 – Interfuncionamiento de los sistemas de señalización. Recomendaciones Q.601 a Q.699 (Comisión de Estudio XI).

FASCÍCULO VI.7 – Especificaciones del sistema de señalización N.° 7. Recomendaciones Q.700 a Q.716 (Comisión de Estudio XI).

FASCÍCULO VI.8 – Especificaciones del sistema de señalización N.° 7. Recomendaciones Q.721 a Q.766 (Comisión de Estudio XI).

FASCÍCULO VI.9 – Especificaciones del sistema de señalización N.° 7. Recomendaciones Q.771 a Q.795 (Comisión de Estudio XI).

FASCÍCULO VI.10 – Sistema de señalización digital de abonado N.° 1 (SDA 1), capa enlace de datos. Recomendaciones Q.920 a Q.921 (Comisión de Estudio XI).

FASCÍCULO VI.11 – Sistema de señalización digital de abonado N.º 1 (SDA 1), capa red, gestión usuario-red. Recomendaciones Q.930 a Q.940 (Comisión de Estudio XI).

FASCÍCULO VI.12 – Red móvil terrestre pública, interfuncionamiento con RDSI y RTPC. Recomendaciones Q.1000 a Q.1032 (Comisión de Estudio XI).

FASCÍCULO VI.13 – Red móvil terrestre pública. Parte aplicación móvil e interfaces. Recomendaciones Q.1051 a Q.1063 (Comisión de Estudio XI).

FASCÍCULO VI.14 – Interfuncionamiento con sistemas móviles por satélite. Recomendaciones Q.1100 a Q.1152 (Comisión de Estudio XI).

**Tomo VII**

FASCÍCULO VII.1 – Transmisión telegráfica. Recomendaciones de la serie R. Equipos terminales para los servicios de telegrafía. Recomendaciones de la serie S (Comisión de Estudio IX).

FASCÍCULO VII.2 – Conmutación telegráfica. Recomendaciones de la serie U (Comisión de Estudio IX).

FASCÍCULO VII.3 – Equipo terminal y protocolos para los servicios de telemática. Recomendaciones T.0 a T.63 (Comisión de Estudio VIII).

FASCÍCULO VII.4 – Procedimientos de prueba de conformidad para las Recomendaciones teletex. Recomendación T.64 (Comisión de Estudio VIII).

FASCÍCULO VII.5 – Equipo terminal y protocolos para servicios de telemática. Recomendaciones T.65 a T.101 y T.150 a T.390 (Comisión de Estudio VIII).

FASCÍCULO VII.6 – Equipo terminal y protocolos para servicios de telemática. Recomendaciones T.400 a T.418 (Comisión de Estudio VIII).

FASCÍCULO VII.7 – Equipo terminal y protocolos para servicios de telemática. Recomendaciones T.431 a T.564 (Comisión de Estudio VIII).

**Tomo VIII**

FASCÍCULO VIII.1 – Comunicación de datos por la red telefónica. Recomendaciones de la serie V (Comisión de Estudio XVII).

FASCÍCULO VIII.2 – Redes de comunicación de datos: servicios y facilidades, interfaces. Recomendaciones X.1 a X.32 (Comisión de Estudio VII).

FASCÍCULO VIII.3 – Redes de comunicación de datos: transmisión, señalización y conmutación, aspectos de red, mantenimiento, disposiciones administrativas. Recomendaciones X.40 a X.181 (Comisión de Estudio VII).

FASCÍCULO VIII.4 – Redes de comunicación de datos: Interconexión de sistemas abiertos (ISA) – Modelo y notación, definición del servicio. Recomendaciones X.200 a X.219 (Comisión de Estudio VII).

FASCÍCULO VIII.5 – Redes de comunicación de datos: Interconexión de sistemas abiertos (ISA) – Especificación de protocolos, pruebas de conformidad. Recomendaciones X.220 a X.290 (Comisión de Estudio VII).

FASCÍCULO VIII.6 – Redes de comunicación de datos: Interfuncionamiento entre redes, sistemas móviles de transmisión de datos, gestión interredes. Recomendaciones X.300 a X.370 (Comisión de Estudio VII).

FASCÍCULO VIII.7 – Redes de comunicación de datos: Sistemas de tratamiento de mensajes. Recomendaciones X.400 a X.420 (Comisión de Estudio VII).

FASCÍCULO VIII.8 – Redes de comunicación de datos: La guía. Recomendaciones X.500 a X.521 (Comisión de Estudio VII).

**Tomo IX** – Protección contra las perturbaciones. Recomendaciones de la serie K (Comisión de Estudio V) – Construcción, instalación y protección de los cables y otros elementos de planta exterior. Recomendaciones de la serie L (Comisión de Estudio VI).

**Tomo X**

FASCÍCULO X.1 – Lenguaje de especificación y descripción funcionales (LED). Criterios para la utilización de técnicas de descripción formal (TDF). Recomendación Z.100 y anexos A, B, C y E, Recomendación Z.110 (Comisión de Estudio X).

FASCÍCULO X.2 – Anexo D a la Recomendación Z.100: Directrices para el usuario del LED (Comisión de Estudio X).

FASCÍCULO X.3 – Anexo F.1 a la Recomendación Z.100: Definición formal del LED. Introducción (Comisión de Estudio X).

FASCÍCULO X.4 – Anexo F.2 a la Recomendación Z.100: Definición formal del LED. Semántica estática (Comisión de Estudio X).

FASCÍCULO X.5 – Anexo F.3 a la Recomendación Z.100: Definición formal del LED. Semántica dinámica (Comisión de Estudio X).

FASCÍCULO X.6 – Lenguaje de alto nivel del CCITT (CHILL). Recomendación Z.200 (Comisión de Estudio X).

FASCÍCULO X.7 – Lenguaje hombre-máquina (LHM). Recomendaciones Z.301 a Z.341 (Comisión de Estudio X).

## INDICE DEL FASCICULO VIII.8 DEL LIBRO AZUL



## NOTAS PRELIMINARES

1 Las Cuestiones asignadas a cada Comisión de Estudio para el periodo de estudios 1989-1992 figuran en la contribución N.º 1 de dicha Comisión.

2 En este fascículo, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una Administración de telecomunicaciones como una empresa privada de explotación de telecomunicaciones reconocida.

3 Los términos anexo y apéndice a las Recomendaciones de la serie X deberán interpretarse como sigue (salvo indicación en contrario):

- el *anexo* a una Recomendación forma parte integrante de la misma;
- el *apéndice* a una Recomendación no forma parte integrante de la misma y tiene solamente por objeto proporcionar explicaciones o informaciones complementarias.

4 Las Recomendaciones de la serie X contenidas en este fascículo fueron preparadas conjuntamente, en colaboración con la ISO/CEI. En el cuadro que sigue se facilitan las referencias mutuas entre estas Recomendaciones y las normas ISO/CEI correspondientes.

| Recomendación del CCITT | Norma o Informe técnico ISO/CEI |
|---|---|
| X.500 | ISO 9594-1, Information processing systems - Open Systems Interconnection - The Directory - Part 1: Overview of concepts, models and service [a] |
| X.501 | ISO 9594-2, Information processing systems - Open Systems Interconnection - The Directory - Part 2: Models [a] |
| X.509 | ISO 9594-8, Information processing systems - Open Systems Interconnection - The Directory - Part 8: Authentication framework [a] |
| X.511 | ISO 9594-3, Information processing systems - Open Systems Interconnection - The Directory - Part 3: Abstract service definition [a] |
| X.518 | ISO 9594-4, Information processing systems - Open Systems Interconnection - The Directory - Part 4: Procedures for distributed operations [a] |
| X.519 | ISO 9594-5, Information processing systems - Open Systems Interconnection - The Directory - Part 5: Protocol specifications [a] |
| X.520 | ISO 9594-6, Information processing systems - Open Systems Interconnection - The Directory - Part 6: Selected attribute types [a] |
| X.521 | ISO 9594-7, Information processing systems - Open Systems Interconnection - The Directory - Part 7: Selected object classes [a] |

[a] Actualmente a nivel de Proyecto de Norma Internacional (PNI).

FASCÍCULO VIII.8

**Recomendaciones X.500 a X.521**

# REDES DE COMUNICACIÓN DE DATOS: LA GUÍA

PAGE INTENTIONALLY LEFT BLANK

PAGE LAISSEE EN BLANC INTENTIONNELLEMENT

# LA GUIA - VISION DE CONJUNTO DE CONCEPTOS, MODELOS Y SERVICIOS [1)]

*(Melbourne, 1988)*

INDICE



---

1) La Recomendación X.500 y la norma ISO 9594-1, The Directory - Overview of Concepts, Models and Services (La guía - Visión de conjunto de conceptos, modelos y servicios) se elaboraron en estrecha colaboración y están técnicamente alineadas.

## 0 Introducción

0.1 Este documento, junto con los otros de la misma serie, ha sido elaborado para facilitar la interconexión de sistemas de procesamiento de información para la prestación de servicios de guía. El conjunto de todos estos sistemas, junto con la información de guía que contienen, puede considerarse como un todo integrado, denominado *la guía*. La información contenida en la guía denominada en forma colectiva base de información de la guía (BIG), se utiliza típicamente para facilitar la comunicación entre, con o sobre objetos tales como entidades de aplicación, personas, terminales, y listas de distribución.

0.2 La guía desempeña un papel importante en la interconexión de sistemas abiertos (ISA), cuyo propósito es permitir, con un mínimo de acuerdos técnicos fuera de las propias normas de interconexión, la interconexión de sistemas de procesamiento de información:

- de diferentes fabricantes;
- sometidos a gestiones diferentes;
- de diferentes grados de complejidad; y
- de diferentes fechas de construcción.

0.3 Esta Recomendación presenta y modela los conceptos de la guía y de la BIG y expone los servicios y capacidades que las mismas proporcionan. Otras Recomendaciones utilizan estos modelos para definir el servicio abstracto prestado por la guía y para especificar los protocolos mediante los cuales se puede obtener o propagar este servicio.

## 1 Alcance y campo de aplicación

1.1 La guía proporciona las capacidades de guía que necesitan las aplicaciones ISA, los procesos de gestión ISA, otras entidades de capa ISA y los servicios de telecomunicaciones. Entre las capacidades que proporciona están la de "denominación cómoda para el usuario", lo que permite aludir a los objetos por nombres adecuados para uso de usuarios humanos (si bien no todos los objetos necesitan contar con nombres cómodos para el usuario), y la de "correspondencia de nombre con dirección" lo que permite una vinculación dinámica entre los objetos y su ubicación. Esta última capacidad permite, por ejemplo, la "autoconfiguración" de las redes ISA, en el sentido de que la adición, supresión y cambios de ubicación de objeto no afecta a la operación de la red ISA.

1.2 La guía no tiene el propósito de ser un sistema de base de datos para fines generales, si bien puede construirse a base de dichos sistemas. Por ejemplo, se supone, como es típico en las guías de comunicaciones, que existe una frecuencia bastante superior de "interrogaciones" (o "indagaciones") que de actualizaciones. La tasa de actualizaciones se supone estará determinada por la dinámica de las personas y organizaciones más bien que, por ejemplo, la dinámica de las redes. Tampoco son necesarias actualizaciones globales instantáneas: las condiciones transitorias, en que se dispone tanto de la versión antigua como nueva de una misma información, son perfectamente aceptables.

1.3 Es característico de la guía que, salvo si existen diferentes derechos de acceso o actualizaciones no propagadas, los resultados de las interrogaciones de la guía no dependerán de la identidad o ubicación del interrogador. Esto hace que la guía no sea adecuada para ciertas aplicaciones de telecomunicaciones, por ejemplo, algunos tipos de encaminamiento.

## 2 Referencias

Recomendación X.200 - Modelo de referencia básico de interconexión de sistemas abiertos para aplicaciones del CCITT

Recomendación X.208 - Interconexión de sistemas abiertos - Especificación de la notación de sintaxis abstracta uno (NSA.1)

Recomendación X.501 - La guía - Modelos

Recomendación X.509 - La guía - Marco de autenticación

Recomendación X.511 - La guía - Definición de servicio abstracto

Recomendación X.518 - La guía - Procedimientos para operación distribuida

Recomendación X.519 - La guía - Especificaciones de protocolos

Recomendación X.520 - La guía - Tipos de atributo seleccionados

Recomendación X.521 - La guía - Clases de objeto seleccionadas

Recomendación X.219 - Operaciones a distancia: Modelo, notación y definición de servicio

Recomendación X.229 - Operaciones a distancia: Especificación de protocolo.

## 3 Definiciones

En las definiciones contenidas en esta sección se utilizan las abreviaturas definidas en el § 4.

### 3.1 *Definiciones del modelo de referencia ISA*

Esta Recomendación se basa en los conceptos expuestos en la Recomendación X.200 y utiliza los siguientes términos definidos en la misma:

a) *entidad de aplicación*;
b) *capa de aplicación*;
c) *proceso de aplicación*;
d) *unidad de datos de protocolo de aplicación*;
e) *elemento de servicio de aplicación.*

### 3.2 *Definiciones básicas de la guía*

a) La *guía*: una colección de sistemas abiertos que cooperan para proporcionar servicios de guía.

b) *Base de información de la guía* (*BIG*): el conjunto de informaciones gobernado por la guía.

c) *Usuario* (*de la guía*): el usuario final de la guía, es decir, la entidad o persona que gana acceso a la guía.

### 3.3 *Definiciones del modelo de la guía*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.501:

a) *dominio de gestión de la guía de administración*;
b) *alias*;
c) *atributo*;
d) *tipo de atributo*;
e) *valor de atributo*;
f) *árbol de información de la guía* (*AIG*);
g) *dominio de gestión de la guía* (*DGG*);
h) *agente de sistema de la guía* (*ASG*);
i) *agente de usuario de la guía* (*AUG*);
j) *nombre distinguido*;
k) *asiento*;
l) *nombre*;
m) *objeto* (*de interés*);
n) *dominio de gestión privado de la guía*;
o) *nombre distinguido relativo*;
p) *raíz*;
q) *esquema*;
r) *objeto subordinado*;
s) *asiento superior*;

t) *objeto superior*;

u) *árbol.*

3.4 *Definiciones relativas a la operación distribuida*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.518:

a) *concatenación*;

b) *difusión*;

c) *referimiento.*

## 4 Abreviaturas

| | |
|---|---|
| AIG | Arbol de información de la guía |
| ASG | Agente de sistema de la guía |
| AUG | Agente de usuario de la guía |
| BIG | Base de información de la guía |
| DGGAD | Dominio de gestión de la guía de administración |
| DGG | Dominio de gestión de la guía |
| DGGPR | Dominio de gestión de la guía privado |
| ISA | Interconexión de sistemas abiertos |
| NDR | Nombre distinguido relativo |
| PAG | Protocolo de acceso a la guía |
| PSG | Protocolo de sistema de guía. |

## 5 Visión de conjunto de la guía

5.1 La *guía* es una colección de sistemas abiertos que cooperan para mantener una base de datos lógica de información sobre un conjunto de objetos del mundo real. Los *usuarios* de la guía, que son personas y programas de computador, pueden leer o modificar la información o parte de ella, a condición de que estén autorizados a hacerlo. Cada usuario está representado, al ganar acceso a la guía, por un agente de usuario de guía (AUG), que se considera como un proceso de aplicación. Estos conceptos se ilustran en la figura 1/X.500.

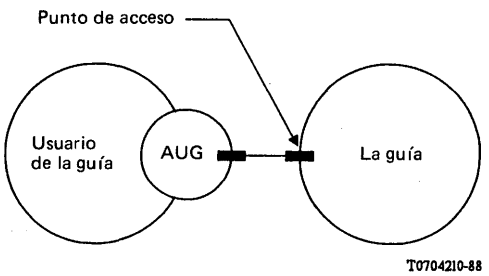



FIGURA 1/X.500

**Acceso a la guía**

*Nota* - Esta serie de Recomendaciones se refiere a la guía en singular, y refleja la intención de crear, por medio de un espacio de nombre, único, unificado, una guía lógica compuesta de muchos sistemas que sirven a muchas aplicaciones. El hecho de que estos sistemas decidan o no interfuncionar dependerá de las necesidades de las aplicaciones soportadas por dichos sistemas. Es posible que aplicaciones referentes a mundos de objetos no intersectantes no tengan esa necesidad. El espacio de nombre único facilitará ese interfuncionamiento, en caso de que las necesidades cambiasen.

5.2 La información contenida en la guía se conoce colectivamente como *base de información de la guía* (BIG). El § 6 analiza su estructura.

5.3 La guía proporciona a sus usuarios un conjunto bien definido de capacidades de acceso, conocido como el servicio abstracto de la guía. Este servicio, que se describe en el § 7, proporciona una capacidad simple de recuperación y modificación. Esta puede combinarse con funciones AUG locales para proporcionar las capacidades requeridas por los usuarios finales.

5.4 Es posible que la guía sea distribuida, tal vez muy distribuida, extensa, según líneas tanto funcionales como organizacionales. En el § 8 se analizan los modelos correspondientes de la guía. Estos han sido elaborados con el objeto de establecer un marco de cooperación de los diversos componentes, a fin de ofrecer un todo integrado.

5.5 El suministro y consumo de los servicios de la guía requiere que los usuarios (en realidad los AUG) y los diversos componentes funcionales de la guía cooperen entre sí. En muchos casos esto requerirá una cooperación entre procesos de aplicación de diferentes sistemas abiertos, lo que a su vez exige protocolos de aplicación normalizados, que se analizan en el § 9, para regular esta cooperación.

5.6 La guía está concebida de manera que soporte múltiples aplicaciones, entre una amplia gama de posibilidades. La naturaleza de las aplicaciones soportadas determinará qué objetos aparecerán en la guía, cuáles usuarios tendrán acceso a la información, y qué tipos de acceso podrán realizarse. Las aplicaciones pueden ser específicas, como por ejemplo el suministro de listas de distribución para el correo electrónico, o genéricas, como la aplicación "guía de comunicaciones interpersonales". La guía ofrece la posibilidad de explotar los aspectos comunes a las diversas aplicaciones:

- un mismo objeto puede concernir a más de una aplicación y tal vez incluso una misma información sobre un mismo objeto.

Para permitir esto se definen cierto número de clases de objeto y de tipos de atributo, que serán útiles para toda una gama de aplicaciones. Estas definiciones se encuentran en las Recomendaciones X.520 y X.521:

- algunos esquemas de utilización de la guía serán comunes a toda una gama de aplicaciones; este aspecto se analiza más a fondo en el anexo A.

## 6 La base de información de la guía (BIG)

*Nota* - La BIG y su estructura se definen en la Recomendación X.501.

6.1 La BIG está formada por información sobre objetos. Está compuesta de *asientos* (de la *guía*), cada uno de los cuales consiste en una colección de informaciones sobre un objeto. Cada asiento está formado de *atributos*, cada uno con un tipo y uno o más valores. Los tipos de atributo presentes en un determinado asiento dependen de la *clase* de objeto descrita por dicho asiento.

6.2 Los asientos de la BIG están organizados en forma de árbol, el arbol de información de la guía (AIG), donde los vértices representan los asientos. Los asientos más elevados del árbol (más cercanos a la raíz) representarán a menudo objetos tales como países u organizaciones, mientras los asientos situados más abajo en el árbol representarán personas o procesos de aplicación.

*Nota* - Los servicios definidos en esta Recomendación operan únicamente en una estructura en forma de árbol. Esta Recomendación no excluye la existencia futura de otras estructuras (según las necesidades).

6.3 Cada asiento tiene un *nombre distinguido*, que identifica dicho asiento en forma única e inequívoca. Estas propiedades del nombre distinguido se derivan de la estructura arboriforme de la información. El nombre distinguido de un asiento está compuesto del nombre distinguido de su asiento superior, junto con los valores de atributo especialmente nominados (los valores *distinguidos*) del asiento.

6.4 Algunos de los asientos en las hojas del árbol son asientos de alias, mientras que otros asientos son asientos de objeto. Los asientos de alias apuntan a asientos de objeto, y proporcionan la base de nombres alternativos para los objetos correspondientes.

6.5 La guía aplica un conjunto de reglas para garantizar que la BIG permanezca bien formada ante las modificaciones que surjan con el tiempo. Estas reglas, conocidas por el nombre de *esquema de la guía*, evitan que los asientos tengan tipos de atributo incorrectos para su clase de objeto, que los valores de atributo tengan una forma errónea para el tipo de atributo, e incluso que los asientos tengan asientos subordinados de una clase errónea.

6.6 La figura 2/X.500 ilustra los conceptos antes mencionados del AIG y sus componentes.

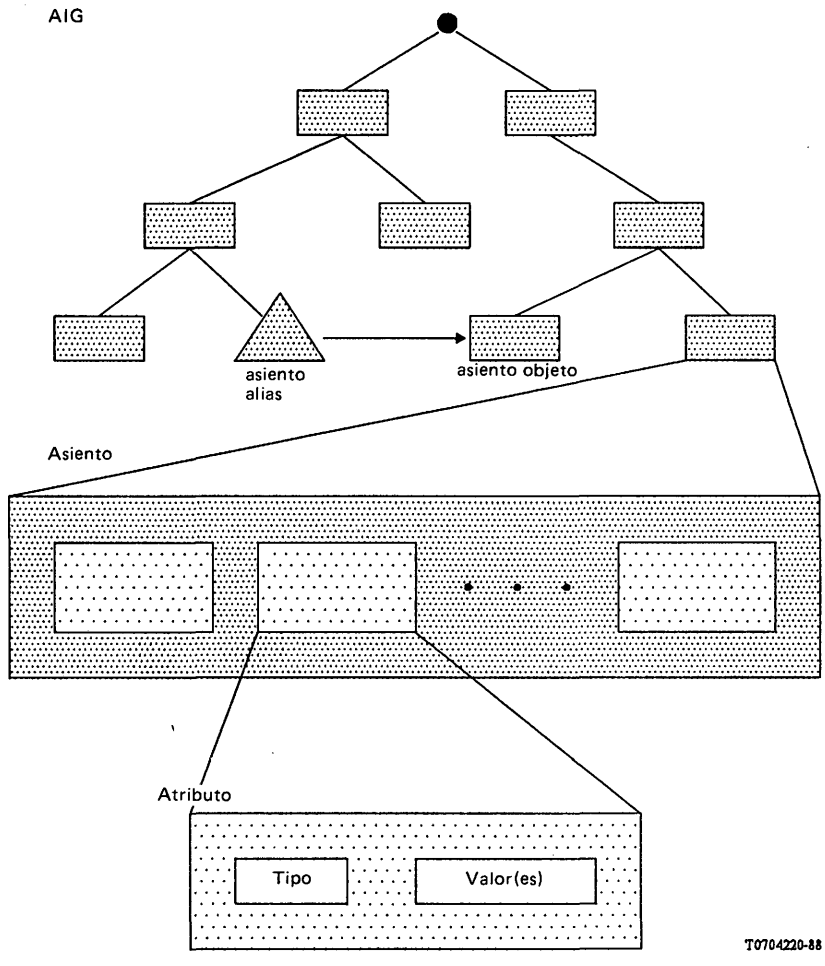



FIGURA 2/X.500

**Estructura del AIG y de los asientos**

6.7 La figura 3/X.500 presenta un ejemplo ficticio de un AIG. El árbol proporciona ejemplos de algunas de las clases de atributo utilizadas para identificar diferentes objetos. Por ejemplo, el nombre:

{C = GB, L = Winslow, O = Servicios gráficos, CN = Impresora láser}

identifica la entidad de aplicación "impresora láser" que tiene, en su nombre distinguido, el atributo geográfico de Localidad. El abonado residencial John Jones, cuyo nombre es:

{C = GB, L = Winslow, CN = John Jones}

tiene el mismo atributo geográfico en su nombre distinguido.

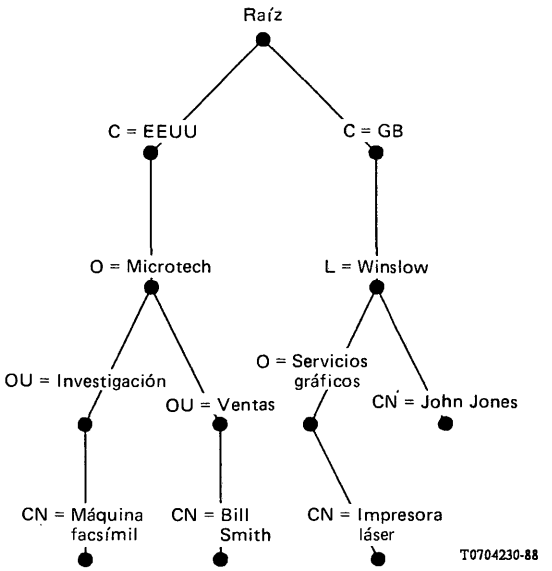



FIGURA 3/X.500

**Un árbol de información de la guía hipotético**

6.8 El crecimiento y forma del AIG, la definición del esquema de la guía, y la selección de nombres distinguidos para los asientos conforme se añaden, es la responsabilidad de diversas autoridades cuyas relaciones jerárquicas se reflejan en la forma del árbol. Por ejemplo, las autoridades deben garantizar que todos los asientos bajo su jurisdicción tengan nombres distinguidos inequívocos, manejando cuidadosamente los valores y tipos de atributo que aparecen en dichos nombres. La responsabilidad se transmite hacia abajo a lo largo del árbol, pasando de autoridades superiores a autoridades subordinadas, ejerciéndose el control por medio del esquema.

## 7 El servicio de guía

*Nota* - La definición del servicio abstracto de guía figura en la Recomendación X.511.

### 7.1 *Introducción*

7.1.1 En esta sección se presenta una descripción general del servicio proporcionado a los usuarios, representados por sus AUG. Todos los servicios son proporcionados por la guía en respuesta a solicitudes de los AUG. Existen peticiones que permiten interrogar la guía, como se describe en el § 7.3, y otras para modificarla, como se describe en el § 7.4. Además, las peticiones de servicio pueden calificarse como se describe en el § 7.2. La guía siempre informa del resultado de cada petición que se le hace. La forma del resultado normal es peculiar de la petición, y resulta evidente a partir de la descripción de la petición. La mayoría de los resultados anormales son comunes a varias peticiones. Las posibilidades se describen en el § 7.5.

7.1.2 Varios aspectos del eventual servicio de guía no están previstos actualmente en las normas especificadas en esta serie de Recomendaciones. Por lo tanto, las capacidades correspondientes deberán proporcionarse como función local hasta que haya una solución normalizada. Estas capacidades incluyen:

- adición y supresión (borrado) de los asientos que se desee eliminar, lo que permite crear una guía distribuida;
- la gestión del control de acceso (es decir, otorgar o retirar, a un usuario específico, el permiso de ganar acceso a una determinada información);
- la gestión del esquema de la guía;
- la gestión de la información "de conocimiento";
- la réplica de partes del BIG.

*Nota* - Esta lista no es exhaustiva.

7.1.3 La guía garantiza que los cambios de la BIG, ya sea como consecuencia de la petición de un servicio de guía o por otro medio (local), den como resultado una BIG que siga obedeciendo las reglas del esquema de la guía.

7.1.4 Un usuario y la guía están vinculados por un periodo de tiempo en un punto de acceso a la guía. En el momento de la vinculación, el usuario y la guía verifican, facultativamente, sus identidades respectivas.

7.2 *Calificación del servicio*

7.2.1 *Controles de servicio*

Se pueden aplicar una serie de medios de control (denominados brevemente controles) a las diversas peticiones de servicio, principalmente con el objeto de que el usuario pueda imponer límites al uso de recursos, que la guía no debe sobrepasar. Se prevén controles, entre otras cosas, del periodo de tiempo, la magnitud de los resultados, el alcance de la búsqueda, o modos de interacción y la prioridad de la petición.

7.2.2 *Parámetros de seguridad*

Cada petición puede ir acompañada de información de soporte de mecanismos de seguridad para proteger la información de la guía. Dicha información puede incluir la petición del usuario de diversos tipos de protección, una firma (signatura) digital de la petición, junto con la información necesaria para que la parte legítimamente interesada pueda verificar la firma.

7.2.3 *Filtros*

Cierto número de peticiones cuyos resultados implican información procedente de o relativa a cierto número de asientos pueden llevar un filtro. El filtro expresa una o más condiciones que el asiento debe satisfacer para poder ser devuelto como parte del resultado. Esto permite reducir el conjunto de asientos devueltos a los que vienen al caso únicamente.

7.3 *Interrogación de la guía*

7.3.1 *Lectura*

Una petición de lectura va dirigida a un asiento específico y provoca la devolución de los valores de algunos o de todos los atributos de dicho asiento. Cuando sólo deben devolverse algunos de los atributos, el AUG suministra la lista de tipos de atributo que interesan.

7.3.2 *Comparación*

Una petición de comparación va dirigida a un atributo particular de un asiento específico y hace que la guía verifique si el valor suministrado concuerda con el valor de dicho atributo.

*Nota* - Por ejemplo, esto puede utilizarse para realizar verificaciones de contraseñas; la contraseña, registrada en la guía, puede ser inaccesible para lectura pero accesible para comparación.

7.3.3 *Listado*

Una petición de listado hace que la guía retorne la lista de subordinados inmediatos de un determinado asiento del AIG.

7.3.4 *Búsqueda*

Una petición de búsqueda hace que la guía retorne información de todos los asientos dentro de cierta porción del AIG que satisfacen cierto filtro. La información retornada de cada asiento consiste en algunos o todos los atributos de dicho asiento, como en lectura.

7.3.5 *Abandono*

Una petición de abandono, aplicada a una petición de interrogación pendiente, informa a la guía que el AUG ya no está interesado en que se atienda la petición. La guía puede, por ejemplo, dejar de procesar la petición, y descartar cualquier resultado que haya logrado hasta ese momento.

7.4 *Modificación de la guía*

7.4.1 *Adición de asiento*

Una petición de adición de asiento hace que se añada al AIG un asiento (un asiento de objeto o un asiento de alias) que constituirá una nueva hoja.

*Nota* - En su forma actual, este servicio está destinado a ser utilizado con el fin de añadir asientos que permanecerán como hojas, tales como asientos para personas o entidades de aplicación, y no para añadir subárboles completos por aplicaciones repetidas de este servicio. Se prevé que el servicio será mejorado en el futuro para atender el caso más general.

7.4.2 *Supresión de asiento*

Como resultado de una petición de supresión de asiento, un asiento constitutivo de hoja es eliminado del AIG.

*Nota* - Al igual que con el servicio de adición de asiento, este servicio está actualmente destinado a aplicarse a asientos constitutivos de "hoja verdadera", y será mejorado en un futuro para atender el caso general.

7.4.3 *Modificación de asiento*

Como resultado de una petición de modificación de asiento, la guía ejecuta una secuencia de cambios en un asiento específico. Se hacen todos los cambios o ninguno de ellos, y la BIG queda siempre en un estado conforme al esquema. Los cambios permitidos son la adición, eliminación o reemplazo de atributos o de valores de atributo.

7.4.4 *Modificación de nombre distinguido relativo*

Como resultado de una petición de modificación de nombre distinguido relativo (NDR), el nombre distinguido de un asiento constitutivo de hoja (sea éste un asiento de objeto o un asiento de alias) en el AIG será modificado por la designación de valores de atributo diferentes de nombre distinguido.

7.5 *Otros resultados*

7.5.1 *Errores*

Cualquier servicio puede fallar, debido por ejemplo a problemas relacionados con los parámetros suministrados por el usuario, en cuyo caso se informa de un error. Siempre que sea posible, con la indicación del error se retorna información para facilitar la solución del problema. Sin embargo, en general, la guía únicamente comunica el primer error que encuentra. Además del ejemplo antes mencionado sobre los problemas que puedan plantear los parámetros suministrados por el usuario (especialmente nombres no válidos de asientos o tipos de atributo no válidos) también pueden surgir errores a causa de violaciones de la política de seguridad, las reglas del esquema y los controles del servicio.

7.5.2 *Reenvíos*

Un servicio puede fallar porque el punto de acceso específico al que está vinculado el AUG no es el más adecuado para atender la petición, por ejemplo, debido a que la información afectada por la petición esté (lógicamente) lejos de ese punto de acceso. En este caso, la guía puede efectuar un reenvío que indique otro punto de acceso en el que el AUG puede presentar su petición.

*Nota* - La guía y el AUG pueden tener una preferencia respecto al uso de reenvíos o a la *concatenación* de las peticiones (véase el § 8.3.3.2). El AUG puede expresar su preferencia por medio de controles del servicio. La guía toma la decisión final en cuanto al método que se va a utilizar.

## 8 La guía distribuida

*Nota* - Los modelos de la guía están definidos en la Recomendación X.501 y los procedimientos para la operación distribuida de la guía se especifican en la Recomendación X.518.

8.1 *Modelo funcional*

En la figura 4/X.500 se muestra el modelo funcional de la guía.

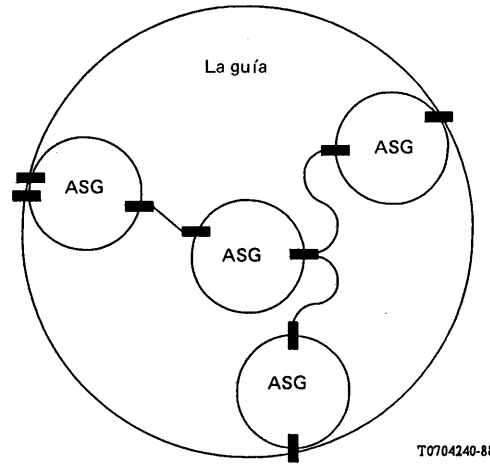



FIGURA 4/X.500

**Modelo funcional de la guía**

Un *agente de sistema de la guía* (ASG) es un proceso de aplicación ISA que forma parte de la guía y cuyo papel es proporcionar a los AUG y/u otros ASG el acceso a la BIG. Un ASG puede utilizar información almacenada en su base de datos local o interactuar con otros ASG para atender las peticiones. Como otra posibilidad, el ASG puede remitir un peticionario a otro ASG que pueda ayudar a atender la petición. Las bases de datos locales dependen enteramente de la implementación.

8.2 *Modelo organizacional*

8.2.1 Un conjunto de uno o más ASG y cero o más AUG manejados por una misma organización pueden formar un dominio de gestión de guía (DGG). La organización de que se trate puede optar por utilizar o no esta serie de Recomendaciones para gobernar las comunicaciones entre los componentes funcionales dentro del DGG.

8.2.2 Las Recomendaciones posteriores especifican ciertos aspectos del comportamiento de los ASG. Con este propósito, un grupo de ASG dentro de un DGG pueden, a modo de opción de la organización que maneja el DGG, comportarse como un ASG único.

8.2.3 Un DGG puede ser un DGG de administración (DGGAD), o un DGG privado (DGGPR), lo que dependerá de que sea o no operado por una organización de telecomunicaciones del sector público.

*Nota* - Debe reconocerse que la provisión de soporte para sistemas de guía privados por parte de miembros del CCITT corresponde al marco de la reglamentación nacional. Así pues, las posibilidades técnicas descritas pueden ser ofrecidas o no por una Administración que proporcione servicios de guía. La operación y la configuración internas de los DGG privados escapa al alcance de las Recomendaciones previstas del CCITT.

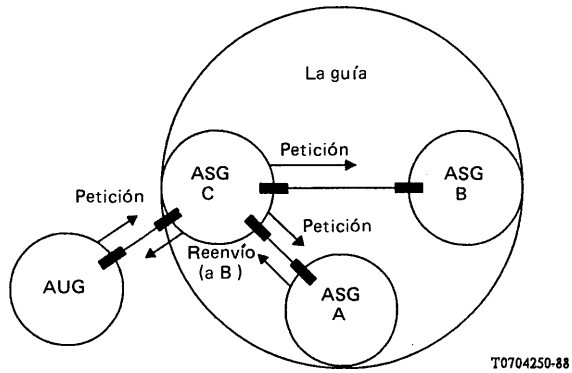
8.3 *Operación del modelo*

8.3.1 El AUG interactúa con la guía comunicando con uno o más ASG. Un AUG no necesita estar vinculado a cualquier ASG específico. Puede interactuar directamente con varios ASG para hacer peticiones. Por razones administrativas, no siempre es posible interactuar directamente con el ASG que tiene que atender la petición, por ejemplo, devolviendo información de la guía. También es posible que el AUG pueda ganar acceso a la guía a través de un solo ASG. Para esto, los ASG necesitarán interactuar entre sí.

8.3.2 El ASG está encargado de atender las peticiones de los AUG, y de obtener la información necesaria cuando no cuenta con ella. Puede cumplir su responsabilidad de obtener la información interactuando con otros ASG en nombre del AUG.

8.3.3 Se han determinado varios casos de tratamiento de peticiones, que se ilustran en las figuras 5/X.500 a 7/X.500 y se describen a continuación.

8.3.3.1 En la figura 5a/X.500, el ASG C recibe un reenvío del ASG A y es responsable de transportar la petición al ASG B (designado en el reenvío del ASG A), o de devolver el reenvío al AUG de origen.




*Nota* - Si ASG C devuelve el reenvío al AUG, la "petición (a B)" no tendrá lugar. De manera similar, si ASG C transporta la petición a ASG B, no devolverá un referimiento al AUG.

FIGURA 5a/X.500

**Reenvíos**

En la figura 5b/X.500, el AUG recibe el reenvío del ASG C, y es responsable, de reemitir la petición directamente a ASG A (designado en el reenvío de ASG A).

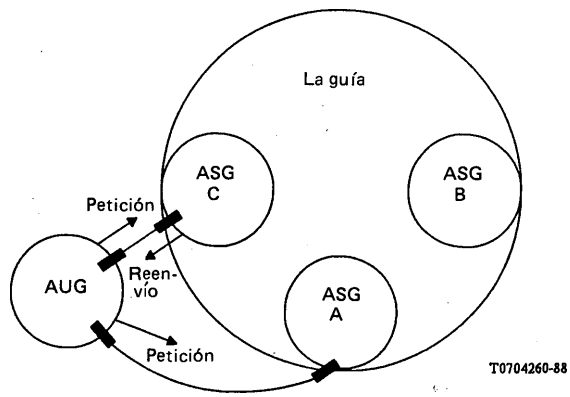



FIGURA 5b/X.500

**Reenvíos**

8.3.3.2 La figura 6/X.500 muestra una concatenación de ASG, en virtud de la cual la petición puede pasar por varios ASG antes de que se retorne la respuesta.

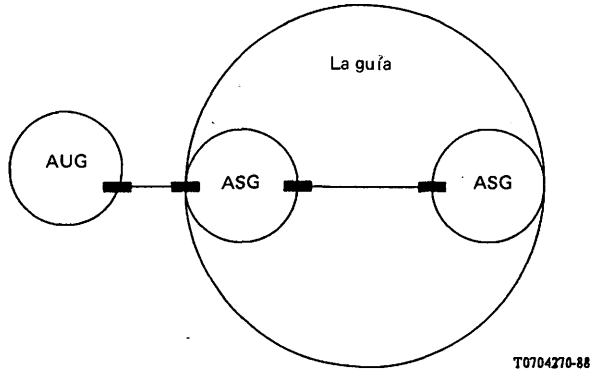



FIGURA 6/X.500

**Concatenación**

8.3.3.3 La figura 7/X.500 muestra la difusión, en virtud de la cual el ASG asociado con el AUG atiende la respuesta, para lo cual reenvía a cada uno de los otros dos ASG una petición idéntica a la recibida.

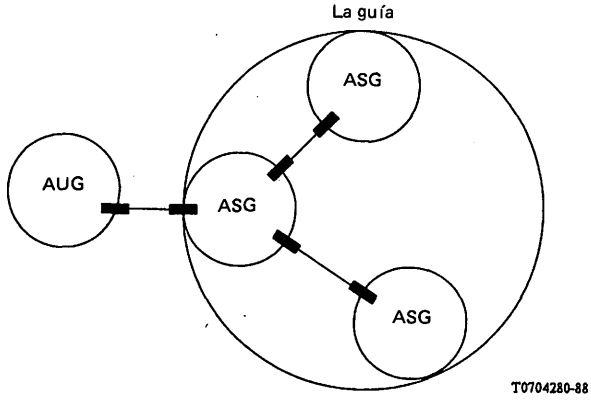



FIGURA 7/X.500

**Difusión**

8.3.4 Cada uno de estos métodos tienen sus propias ventajas. Por ejemplo, el método de la figura 5/X.500 puede utilizarse cuando convenga aligerar la carga del ASG local. En otras circunstancias será necesario un método mixto que combine un conjunto más elaborado de interacciones funcionales, a fin de satisfacer las exigencias del peticionario, como se ilustra en la figura 8/X.500.

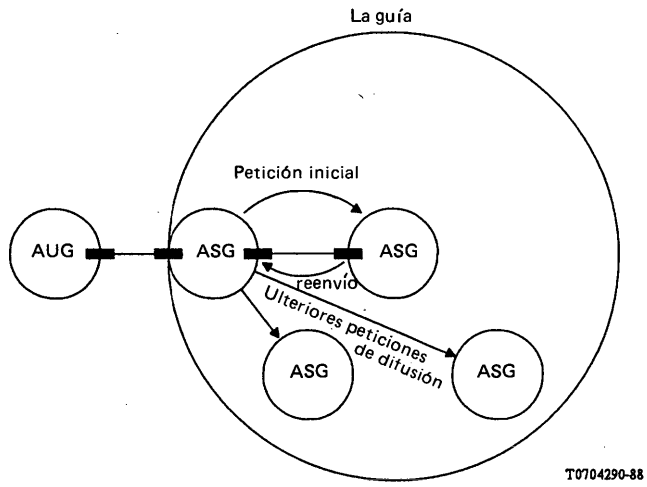



FIGURA 8/X.500

**Método mixto**

## 9 Protocolos de la guía

*Nota* - Los protocolos de la capa de aplicación ISA definidos para permitir la cooperación entre AUG y ASG en diferentes sistemas abiertos se especifican en la Recomendación X.519.

9.1 Hay dos protocolos de la guía:

- el protocolo de acceso a la guía (PAG), que define el intercambio de peticiones y de resultados entre un AUG y un ASG;
- el protocolo de sistema de guía (PSG), que define el intercambio de peticiones y resultados entre dos ASG.

9.2 Cada protocolo está definido por un contexto de aplicación, cada uno de los cuales contiene un conjunto de elementos de protocolo. Por ejemplo, el PAG contiene algunos elementos de protocolo asociados con la interrogación y la modificación de la guía.

9.3 Cada contexto de aplicación está formado de elementos de servicio de aplicación. Estos elementos de servicio de aplicación se definen para el uso del servicio de operaciones a distancia (SOD) de la Recomendación X.219, para estructurar y soportar sus interacciones. Así, los PAG y PSG se definen como conjuntos de operaciones a distancia y errores utilizando la notación SOD.

## ANEXO A

(a la Recomendación X.500)

**Aplicación de la guía**

Este anexo no forma parte integrante de la Recomendación.

### A.1 *El entorno de la guía*

*Nota* - En esta sección, el término "red" se utiliza en su acepción general, para designar el conjunto de sistemas y procesos interconectados relacionados con cualquier servicio de telecomunicaciones, y no únicamente de uno relacionado con la capa de red ISA.

El entorno de la guía, en el que ésta ofrece servicios, es el siguiente:

a) Muchas redes de telecomunicaciones serán de gran envergadura y sufrirán constantes cambios:

   1) objetos de diversos tipos entrarán y saldrán de la red sin ningun aviso, bien por separado o en grupos;

   2) la conectividad de los objetos (especialmente los nodos de la red) cambiará debido a la adición o eliminación de trayectos entre ellos;

   3) diversas características de los objetos, tales como sus direcciones, disponibilidad y ubicación física, pueden cambiar en cualquier momento;

b) si bien la velocidad de cambio, global, es elevada, la vida útil de un objeto determinado no es corta. Por lo común, un objeto participará en comunicaciones con mucha mayor frecuencia que la de cambio de su dirección, disponibilidad, ubicación física, etc.;

c) los objetos que participan en los actuales servicios de telecomunicaciones se identifican por lo general mediante números u otras cadenas de símbolos, seleccionados en función de su facilidad de atribución y procesamiento pero no en función de su facilidad de utilización por los humanos.

### A.2 *Características del servicio de guía*

La necesidad de capacidades de guía obedece:

a) al deseo de aislar (tanto como sea posible) al usuario de la red de los cambios frecuentes que ocurren en ella. Esto puede lograrse creando un "nivel de indirección" ("level of indirection") entre los usuarios y los objetos tratados por ellos. Esto implica que los usuarios se refieran a los objetos dando un nombre y no por ejemplo, una dirección (postal). La guía proporciona el servicio para establecer la necesaria correspondencia;

b) al deseo de presentar una imagen, de la red, que sea más "cómoda para el usuario". Por ejemplo, el uso de alias, la prestación del servicio de "páginas amarillas" (véase el § A.3.5), etc., ayudan a aliviar la labor de encontrar y utilizar la información de la red.

La guía permite a los usuarios obtener una diversidad de informaciones sobre la red y ofrece los medios necesarios para el mantenimiento, la distribución y la seguridad de dicha información.

### A.3 *Patrones de utilización de la guía*

*Nota* - La presente sección se refiere únicamente a la extracción (dícese también, recuperación) de información de la guía; se supone que los servicios de modificación de la guía son utilizados únicamente para el mantenimiento de la BIG en el tiempo, en la forma necesaria para la aplicación.

#### A.3.1 *Introducción*

El servicio de la guía se define en estas normas en base a las peticiones concretas que puede realizar un AUG y a los parámetros de las mismas. Sin embargo, el diseñador de una aplicación, al considerar las exigencias de extracción de información de la guía en dicha aplicación, adoptará probablemente un punto de vista más conforme a su objetivo específico. Por lo tanto, en esta sección se describe una serie de patrones de utilización de alto nivel del servicio de guía que podrían ofrecer interés para muchas aplicaciones.

#### A.3.2 *Consulta*

La consulta directa de la guía es probablemente el tipo de indagación más frecuente en la guía. Para ello, el AUG suministra el nombre distinguido de un objeto junto con un tipo de atributo. La guía devolverá todo valor o valores que correspondían a ese tipo de atributo. Esta es una generalización de la función clásica de la guía, que se obtiene cuando el tipo de atributo solicitado corresponde a un tipo específico de dirección. Los tipos de atributo para los diversos tipos de direcciones están normalizados, lo que incluye la dirección de PASP de ISA, la dirección O/D de tratamiento de mensajes y los números de teléfono y de télex.

La consulta es soportada por el servicio de lectura, que además permite las siguientes generalizaciones adicionales:

- la consulta puede basarse en nombres que no sean el nombre distinguido del objeto, por ejemplo, alias;
- los valores de una serie de tipos de atributos pueden solicitarse en una sola petición, siendo el caso extremo aquel en el que deban devolverse los valores de todos los atributos del asiento en cuestión.

A.3.3 *Denominación cómoda para el usuario*

Se pueden dar a los objetos unos nombres tales que sea máxima la probabilidad de que éstos puedan ser "reconocidos" (o tal vez recordados) por las personas. Los nombres que tienen esta propiedad suelen estar formados de atributos que, de alguna manera, son inherentes al objeto, y no ser "inventados" con dicho propósito. El nombre de un objeto será común a todas las aplicaciones que se refieran al mismo.

A.3.4 *Examen rápido ("hojeado")*

En muchos usos de la guía por el ser humano, tal vez no sea posible para el usuario (o AUG) indicar directamente un nombre, sea este del tipo "cómodo para el usuario", o de otro tipo, del objeto sobre el cual busca información. Sin embargo, tal vez el usuario "lo reconocerá cuando lo vea". La posibilidad de "hojear" la guía permitirá al usuario humano recorrer la BIG para buscar los asientos adecuados.

El examen rápido se logra por una combinación de los servicios de listado y de búsqueda, posiblemente junto con el de lectura (aunque el servicio de búsqueda incluye la capacidad de lectura).

A.3.5 *"Páginas amarillas"*

Hay una diversidad de formas de proporcionar una capacidad del tipo "páginas amarillas". La más sencilla se basa en filtrado, utilizando las aserciones sobre atributos particulares cuyos valores son las categorías (por ejemplo, el tipo de atributo "categoría comercial" definido en la Recomendación X.520). Este método no requiere el establecimiento de ninguna información especial dentro del AIG, salvo para garantizar que están presentes los atributos necesarios. Sin embargo, en el caso general, cuando hay una población, la búsqueda puede resultar costosa ya que el filtrado requiere la generación del conjunto universal a filtrar.

Se puede utilizar otro posible método basado en el establecimiento de subárboles especiales cuyas estructuras de denominación están especialmente concebidas para la búsqueda del tipo "páginas amarillas". La figura A-1/X.500 muestra un ejemplo de un subárbol del tipo "páginas amarillas" en el cual sólo figuran asientos de alias. En realidad, los asientos en los subárboles de "páginas amarillas" pueden ser una combinación de asientos de objeto y de alias, a condición de que sólo exista un asiento de objeto para cada objeto almacenado en la guía.

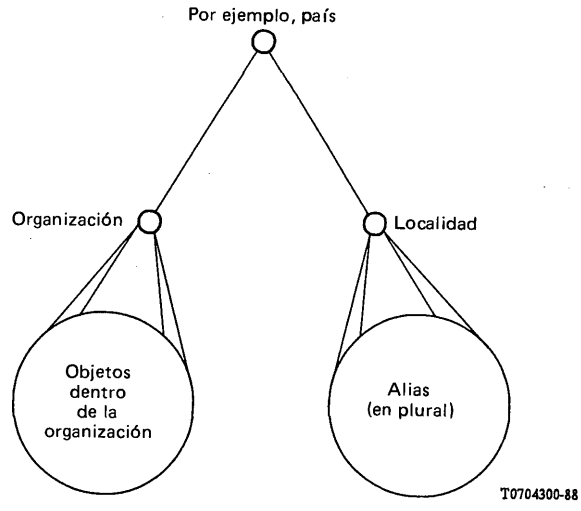



FIGURA A-1/X.500

**Subárbol del tipo "páginas amarillas"**

A.3.6 *Grupos*

Un grupo es un conjunto cuya composición puede cambiar con el tiempo por la adición o supresión explícitas de miembros. El grupo es un objeto, al igual que sus miembros. Se puede pedir a la guía que:

- indique si un objeto dado es o no miembro de un grupo;
- enumere los miembros de un grupo.

Para soportar grupos se hace que el asiento para el grupo contenga un atributo "miembro" de múltiples valores (este tipo de atributo se define en la Recomendación X.520). Las dos capacidades mencionadas pueden entonces realizarse por medio de comparación y lectura respectivamente.

Un miembro de un grupo puede en sí mismo ser un grupo, si esto resulta significativo para la aplicación. Sin embargo, el AUG tendría que crear los servicios de verificación y expansión recursivos necesarios a partir de las versiones no recursivas con que cuenta.

A.3.7 *Autenticación*

Muchas aplicaciones requieren que los objetos participantes ofrezcan cierta prueba de su identidad antes de que se les permita realizar una acción. La guía soporta este proceso de autenticación. (Por otra parte, la propia guía requiere que sus usuarios se autentiquen a sí mismos, para soportar el control de acceso.)

El método de autenticación más directo denominado "autenticación simple", se basa en el uso por la guía de un atributo "contraseña de usuario" en el asiento de cualquier usuario que deseee poder autenticarse a sí mismo ante un servicio. A petición del servicio, la guía confirmará o negará que un valor específico suministrado es realmente la contraseña del usuario. Esto evita que el usuario tenga que poseer una contraseña diferente para cada servicio. En aquellos casos en que se considere inapropiado, el intercambio de contraseñas en un entorno local que utiliza la autenticación simple, la guía podrá, facultativamente, proporcionar medios para proteger esas contraseñas contra una reproducción o uso indebido por una función unidireccional.

El método más complejo, denominado "autenticación fuerte", se basa en la criptografía de claves públicas, en la cual la guía actúa como un depositario de claves públicas de encripción de los usuarios, debidamente protegidas contra las maniobras fraudulentas. El procedimiento que los usuarios pueden seguir para obtener, de la guía, las claves públicas de cada uno de los demás usuarios, y después autenticarse, mutuamente, se describen detalladamente en la Recomendación X.509.

A.4 *Aplicaciones genéricas*

A.4.1 *Introducción*

Cabe imaginar varias aplicaciones genéricas que pueden ser soportadas implícitamente por la guía, es decir, aplicaciones que no son peculiares de ningún servicio en especial de telecomunicaciones. Seguidamente se describen dos de estas aplicaciones: la guía de comunicaciones interpersonales, y la guía de comunicaciones intersistemas (para ISA).

*Nota* - La autenticación, descrita en la sección anterior como un "patrón de acceso" (o "esquema de acceso"), podría igualmente considerarse una aplicación genérica de la guía.

A.4.2 *Comunicaciones interpersonales*

El propósito de esta aplicación es proporcionar a las personas, o a sus agentes, información sobre la forma de comunicarse con otras personas o grupos de personas.

Entran en juego sin duda las siguientes clases de objetos: persona, rol organizacional y grupo. Intervienen también muchas otras clases, probablemente en menor proporción, como: país, organización, unidad organizacional.

Los tipos de atributos que entran en juego, aparte de los utilizados en la denominación, son generalmente atributos de direccionamiento. Típicamente, el asiento para una persona determinada contendrá las direcciones correspondientes a cada uno de los métodos de comunicación por medio de los cuales se puede alcanzar a dicha persona, elegidos de una lista abierta que incluye cuando menos los siguientes: telefonía, correo electrónico, télex, RDSI, entrega física (por ejemplo, el sistema postal), facsímil. En algunos casos, como en el del correo electrónico, el asiento contendrá información adicional, como los tipos de información que puede tratar el equipo del usuario. Si se ha de soportar la autenticación, se necesitarán credenciales y/o contraseñas del usuario.

Los esquemas de denominación utilizados para las diversas clases de objeto deben ser cómodos para el usuario, con los alias adecuados para poder proporcionar nombres alternativos, asegurar la continuidad después del cambio de un nombre, etc.

En esta aplicación se verificarán los siguientes patrones de acceso: consulta, denominación cómoda para el usuario, examen rápido, "páginas amarillas", y grupos. En diversos grados, también se utilizará autenticación.

A.4.3 *Comunicaciones entre sistemas (para ISA)*

De acuerdo con el modelo de referencia de ISA, la ISA requiere dos funciones de guía, a saber, una en la capa de aplicación, que establece la correspondencia de los títulos de aplicación con direcciones de presentación, y otra en la capa de red, que hace establecer la correspondencia de las direcciones PASP con direcciones de PASU (PASU = punto de asociación de subred).

*Nota* - En el resto de esta sección se trata únicamente el caso de la capa de aplicación.

Esta función se realiza consultando la guía si la información requerida para establecer la correspondencia no está disponible localmente.

Los usuarios son entidades de aplicación, y las clases de objeto que interesan son también entidades de aplicación, o subclases de las mismas.

El principal tipo de atributo que interviene, aparte de los utilizados para la denominación, es el de dirección de presentación. Otros tipos de atributos, que no se consideran necesarios para la propia función de guía, podrían soportar la verificación o determinación del tipo de entidad de aplicación, o las listas de contextos de aplicación, sintaxis abstractas, etc. admitidos. Los tipos de atributos relacionados con la autenticación también pueden ser aplicables.

El principal patrón de acceso que se verificará será el de consulta.

**Recomendación X.501**

## LA GUIA - MODELOS [1]

*(Melbourne, 1988)*

INDICE



---

1) La Recomendación X.501 y la norma ISO 9594-2, The Directory - Models, (la guía-modelos) se redactaron en estrecha colaboración y están técnicamente alineadas.

## 0 Introducción

0.1 Este documento, junto con los demás de la serie, ha sido elaborado para facilitar la interconexión de los sistemas de tratamiento de la información con objeto de ofrecer servicios de guía. Un conjunto de tales sistemas, además de la información de guía que contienen, puede considerarse como un todo integrado, llamado la *guía.* la información que contiene la guía, conocida colectivamente como base de información de la guía (BIG), suele utilizarse para facilitar la comunicación entre, con o sobre objetos tales como entidades de aplicación, personas, terminales y listas de distribución.

0.2 La guía desempeña un papel importante en la interconexión de sistemas abiertos, cuya finalidad es permitir, con un mínimo de acuerdo técnico aparte de las normas de interconexión en sí mismas, la interconexión de sistemas de procesamiento de información:

- de diferentes fabricantes;
- sometidos a gestiones diferentes;
- de diferentes grados de complejidad; y
- de diferentes fechas de construcción.

0.3 Esta Recomendación proporciona una serie de modelos diferentes para la guía que sirven de marco a las demás Recomendaciones. Los modelos son el modelo global (funcional); el modelo orgánico; el modelo de seguridad, y el marco de información. Este último describe la manera en la cual la guía organiza la información que posee. Por ejemplo, describe de qué modo la información relativa a objetos se agrupa para constituir a los respectivos asientos de guía y cómo esa información da nombres a objetos.

0.4 En el anexo A se resume la terminología matemática asociada con estructuras de árbol.

0.5 En el anexo B se resume la utilización de los identificadores de objeto ASN.1 en esta serie de Recomendaciones.

0.6 En el anexo C se proporciona el módulo ASN.1 que contiene todas las definiciones asociadas con el marco de información.

0.7 En el anexo D se enumeran alfabéticamente los términos definidos en esta Recomendación.

0.8 En el anexo E se describen algunos criterios que pueden tomarse en consideración al establecer nombres.

0.9 El anexo F contiene las directrices para el control de acceso.

## 1 Alcance y campo de aplicación

1.1 Los modelos definidos en esta Recomendación dan un marco conceptual y terminológico para las demás Recomendaciones que definen diversos aspectos de la guía.

1.2 Los modelos funcional y organizacionales definen posibles formas de distribución de la guía, tanto funcional como administrativa.

1.3 El modelo de seguridad define el marco dentro del cual la guía proporciona dispositivos de seguridad, tal como el control del acceso.

1.4 El marco de información describe la estructura lógica de la BIG. Desde este punto de vista, el hecho de que la guía sea distribuida y no centralizada no es visible. Las demás Recomendaciones de la serie se valen de los conceptos del marco de información. En particular:

a) el servicio que ofrece la guía se describe (en la Recomendación X.511) con arreglo a los conceptos del marco de información, gracias a lo cual el servicio puede guardar una cierta independencia con respecto a la distribución física de la BIG;

b) la operación distribuida de la guía se especifica (en la Recomendación X.518) con el fin de prestar ese servicio y, por ende, mantener esa estructura lógica de información, partiendo de la premisa que la BIG es en realidad altamente distribuida.

## 2 Referencias

Recomendación X.200 - Modelo de referencia de interconexión de sistemas abiertos para aplicaciones del CCITT.

Recomendación X.500 - La guía - Visión de conjunto de conceptos, modelos y servicios.

Recomendación X.509 - La guía - Marco de autenticación.

Recomendación X.511 - La guía - Definición de servicio abstracto.

Recomendación X.518 - La guía - Procedimientos para operación distribuida.

Recomendación X.519 - La guía - Especificaciones de protocolos de acceso y de sistema.

Recomendación X.520 - La guía - Tipos de atributo seleccionados.

Recomendación X.521 - La guía - Clases de objeto seleccionadas.

## 3 Definiciones

Al comienzo de cada cláusula figuran definiciones de los términos pertinentes. Para facilitar las referencias, el anexo D contiene un índice alfabético de estos términos.

## 4 Abreviaturas

| | |
|---|---|
| AIG | Arbol de información de la guía |
| ASG | Agente de sistema de guía |
| AUG | Agente de usuario de guía |
| AVA | Aserción de valor de atributo |
| BIG | Base de información de la guía |
| DGG | Dominio de gestión de la guía |
| DGGAD | Dominio de gestión de la guía de Administración |

DGGPR Dominio de gestión de la guía privado

NDR Nombre distinguido relativo

SECCION 1 - *Modelo de guía*

## 5 Modelo de guía

### 5.1 *Definiciones*

a) *punto de acceso:* punto en el que se obtiene un servicio abstracto;

b) *dominio de gestión de la guía de Administración (DGGAD):* un DGG manejado por una Administración;

*Nota* - El término "Administración" designa a un servicio público de telecomunicaciones u otra organización que ofrezca servicios públicos de telecomunicaciones.

c) *autoridad administrativa:* entidad que ejerce el control administrativo de todos los asientos almacenados en un solo agente de sistema de gestión;

d) *la guía:* un depositario de información acerca de objetos, que proporciona servicios de guía a sus usuarios y permite el acceso a la información;

e) *dominio de gestión de la guía (DGG):* colección de uno o más ASG y cero o más AUG, manejada por una sola organización;

f) *agente de sistema de guía (ASG):* un proceso de aplicación ISA que forma parte de la guía;

g) *usuario (de la guía):* el usuario final de la guía, por ejemplo, la entidad o persona que gana acceso a la misma;

h) *agente de usuario de guía (AUG):* un proceso de aplicación ISA que representa a un usuario que gana acceso a la guía;

*Nota* - Los AUG pueden también facilitar una serie de servicios locales para ayudar a los usuarios a formular preguntas e interpretar las respuestas.

i) *dominio de gestión de la guía privado (DGGPR):* un DGG manejado por una organización que no es una administración.

### 5.2 *La guía y sus usuarios*

5.2.1 Un usuario de la guía (por ejemplo una persona o un proceso de aplicación) obtiene los servicios correspondientes mediante el acceso a la *guía*. Dicho sea de una manera más precisa es un *agente de usuario de guía* (AUG) quien tiene acceso efectivo a la guía e interactúa con la misma para obtener el servicio en nombre de cierto usuario. La guía ofrece uno o más *puntos de acceso* en los que pueden producirse esos accesos. Estos conceptos aparecen ilustrados en la figura 1/X.501.

5.2.2 Los servicios ofrecidos por la guía se definen en la Recomendación X.511.

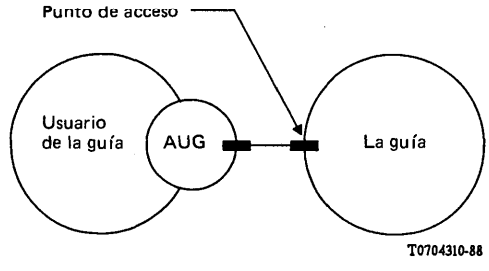



FIGURA 1/X.501

**Acceso a la guía**

5.2.3 La guía es un depositario de información acerca de objetos, y los servicios de guía que proporciona a sus usuarios se refieren a diversos tipos de acceso a esta información. La información se conoce colectivamente como la *base de información de la guía* (*BIG*). En la sección 2 de esta Recomendación se define un modelo de BIG.

5.2.4 Un AUG se manifiesta como un proceso de aplicación. Cada AUG representa precisamente a un usuario de la guía.

*Nota 1* - Algunos sistemas abiertos pueden ofrecer una función AUG centralizada que extrae la información para los verdaderos usuarios (procesos de aplicación, personas, etc.) Esto es transparente para la guía.

*Nota 2* - Las funciones ASG y AUG (véase el § 5.3.1) pueden estar en el mismo sistema abierto y el hacer visibles a uno o más ASG en el entorno ISA como entidades de aplicación es una elección en la realización.

*Nota 3* - Es probable que los AUG presenten un comportamiento y una estructura locales no pertenecientes al marco de Recomendaciones previstas. Por ejemplo, un AUG que representa a un usuario humano de la guía puede proporcionar una serie de servicios (facilidades) locales para ayudar al usuario a formular preguntas e interpretar respuestas.

5.3 *Modelo funcional*

5.3.1 La guía reviste la forma de un conjunto compuesto por uno o más procesos de aplicación conocidos como *agentes de sistema de guía* (*ASG*), cada uno de los cuales proporciona cero, uno o más puntos de acceso. Esto se encuentra ilustrado en la figura 2/X.501. Cuando la guía está compuesta por más de un ASG, se dice que es *distribuida*. Los procedimientos para la operación de la guía cuando ésta es de tipo distribuido se especifican en la Recomendación X.518.

*Nota* - Es probable que los ASG presenten un comportamiento y una estructura locales no pertenecientes al marco de las Recomendaciones previstas. Por ejemplo, un ASG responsable de poseer una parte o toda la información en la BIG lo hará normalmente por medio de una base de datos cuyo interfaz es un asunto local.

5.3.2 Un par específico de procesos de aplicación que deben interactuar para la prestación de servicios de guía (ya sea un AUG y un ASG, o dos ASG) puede estar situado en sistemas abiertos diferentes. Esa interacción se efectúa mediante protocolos de guía ISA, especificados en la Recomendación X.519.

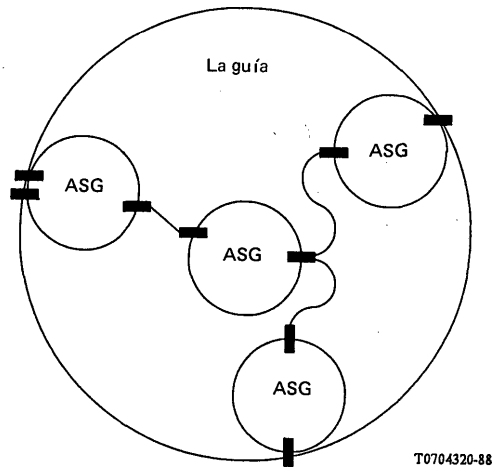



FIGURA 2/X.501

**La guía proporcionada por múltiples ASG**

5.4 *Modelo organizacional*

5.4.1 Un conjunto de uno o más ASG y cero o más AUG manejados por una sola organización puede constituir un *dominio de gestión de la guía* (*DGG*).

*Nota* - La organización que maneja un DGG puede ser una Administración (es decir, una administración pública de telecomunicaciones u otra organización que brinde servicios públicos de telecomunicaciones), en cuyo caso se dirá que el DGG es un DGG de administración (DGGAD); de otro modo se tratará de un DGG privado (DGGPR). Se reconocerá que el soporte a los sistemas privados de guía por parte de miembros del CCITT es de la competencia de las disposiciones reglamentarias nacionales. Por consiguiente, una Administración que proporciona servicios de guía podrá o no ofrecer las posibilidades técnicas descritas. La configuración y el funcionamiento interno de los DGG privados están fuera del ámbito de Recomendaciones previstas del CCITT.

5.4.2 La gestión de un AUG por un DGG implica una responsabilidad constante de prestación de servicios a ese AUG, por ejemplo el DGG tendrá que asegurar el mantenimiento, o en algunos casos la propiedad.

5.4.3 Será facultativo para la organización de que se trate utilizar esta serie de Recomendaciones para gobernar las interacciones que puedan producirse entre AUG y ASG que estén totalmente situadas dentro del DGG.

5.4.4 Cada ASG es administrado por una autoridad administrativa. Esta entidad ejerce el control sobre todos los asientos de objeto y asientos de alias almacenados por dicho ASG. Esto comprende responsabilidades en cuanto al esquema de guía que se está utilizando para orientar la creación y la modificación de asientos (véase el § 9). La estructura y la atribución de nombres compete a una autoridad de denominación [véase el § 8.1 f)], y el cometido de la autoridad administrativa consiste en incorporar efectivamente esas estructuras de denominación en el esquema.

SECCION 2 - *Modelo de información*

## 6 Base de información de la guía

### 6.1 *Definiciones*

a) *asiento de alias:* un asiento de la clase "alias" que contiene información utilizada para proporcionar un nombre alternativo para un objeto;

b) *base de información de la guía (BIG):* el conjunto completo de informaciones a que da acceso la guía, que comprende todas las informaciones que se pueden leer o manipular utilizando las operaciones de la guía;

c) *árbol de información de la guía (AIG):* la BIG considerada como un árbol, cuyos vértices (aparte de la raíz) constituyen los asientos de la guía;

*Nota* - La alternativa se utiliza en lugar de BIG únicamente en contextos donde ofrezca interés la estructura de árbol de la información.

d) *asiento (de la guía):* una parte de la BIG que contiene información sobre un objeto.

e) *superior inmediato* (sustantivo): con relación a un determinado asiento u objeto (debe percibirse claramente del contexto considerado), el asiento u objeto inmediatamente superior;

f) *inmediatamente superior (asiento):* con relación a un asiento determinado - asiento que se halla en el vértice inicial de un arco en el AIG cuyo vértice final es el del asiento en cuestión,

(*objeto*): con relación a un objeto determinado - objeto cuyo asiento de objeto es el superior inmediato de cualquiera de los asientos del segundo objeto;

g) *objeto (de interés):* cualquier cosa de algún ámbito, generalmente el ámbito de las telecomunicaciones y el tratamiento de la información o una parte del mismo, que es identificable (puede ser designada por un nombre), y que ofrece interés el que la BIG mantenga información sobre ella;

h) *clase de objeto:* una familia identificada de objetos (o de objetos concebibles) que comparten algunas características en común;

i) *asiento de objeto:* un asiento que es la colección primaria de información en la BIG sobre un objeto y que, por ende, puede considerarse como representativo del objeto en la BIG;

j) *subclase:* con respecto a una superclase - una clase de objeto derivada de una superclase. Los miembros de la subclase comparten todas las características de otra clase de objeto (la superclase) y otras características que no son poseídas por ninguno de los miembros de esa clase de objeto (la superclase);

k) *subordinado/inferior:* lo opuesto de superior;

l) *superclase:* con relación a una subclase, una clase de objeto de la cual se ha derivado una subclase;

m) *superior:* (aplicado a asiento u objeto) inmediatamente superior, o superior a uno que es inmediatamente superior (recursivamente).

6.2 *Objetos*

6.2.1 La guía tiene por finalidad contener información sobre *objetos de interés* (*objetos*) que existen en algún "mundo" y permitir el acceso a la misma. Un objeto puede ser cualquier cosa de ese mundo que sea identificable (es decir, que pueda ser designada con un nombre).

*Nota 1* - El "mundo" es generalmente el de las telecomunicaciones y el tratamiento de la información o una parte del mismo.

*Nota 2* - Los objetos conocidos para la guía pueden no corresponder exactamente con el conjunto de cosas reales en el mundo. Por ejemplo, una persona del mundo real puede ser considerada como dos objetos diferentes, una persona profesional y una persona residencial por lo que a la guía se refiere. La relación de correspondencia no se define en la presente Recomendación, sino que compete a los usuarios y los proveedores de la guía en el contexto de sus aplicaciones.

6.2.2 El conjunto completo de informaciones a que da acceso la guía se conoce como la *base de información de la guía* (BIG). Se consideran incluidas en la BIG todas las informaciones que pueden ser leídas o manipuladas por las operaciones de la guía.

6.2.3 Una *clase de objeto* es una familia identificada de objetos (u objetos concebibles) que poseen algunas características en común. Todo objeto pertenece al menos a una clase. Una clase de objeto puede ser una *subclase* de otra clase de objeto, en cuyo caso los miembros de la primera clase (la subclase) se consideran también miembros de la segunda (la superclase). Puede haber subclases de subclases, y así sucesivamente hasta un grado arbitrario de subdivisión.

6.3 *Asientos de la guía*

6.3.1 La BIG se compone de *asientos de la guía* (*asientos*) cada uno de los cuales contiene información sobre (que describe) un solo objeto.

6.3.2 Para cada objeto determinado hay exactamente un *asiento de objeto*, que constituye la colección primaria de información sobre el objeto en la BIG acerca de ese objeto. Se dice que el asiento de objeto representa al objeto.

6.3.3 Para cualquier objeto en particular pueden existir, además del asiento de objeto, uno o más *asientos de alias* para ese objeto, que se utilizan para proporcionar nombres alternativos (véase el § 8.5).

6.3.4 La estructura de los asientos de la guía se representa en la figura 3/X.501, y se describe en el § 7.2.

6.3.5 Cada asiento contiene una indicación de la clase de objeto y de las superclases de dicha clase de objeto con las cuales el asiento está asociado. En el caso de un asiento de objeto, éste indica la clase o clases a que pertenece el objeto. En el caso de un asiento de alias, éste indica, mediante una clase de objeto especial, "alias" (definida en el § 9.4.8.2), que se trata en realidad de un asiento de alias, y puede también señalar a qué subclase(s) de la clase de objeto alias pertenece el asiento.

6.4 *El árbol de información de la guía (AIG)*

6.4.1 Para cumplir los requisitos de la distribución y gestión de una BIG de gran amplitud potencial, y a fin de denominar a los objetos sin ninguna ambigüedad (véase § 8), y de hallar los asientos correspondientes, deberá descartarse por poco viable una estructura plana de asientos. Por consiguiente, se podrá aprovechar la relación jerárquica que suele darse entre los objetos (por ejemplo, una persona trabaja en un departamento, que pertenece a una organización, cuya sede se halla en un país determinado), disponiendo los asientos en un árbol, conocido por el *árbol de información de la guía* (*AIG*).

*Nota* - El anexo A contiene una introducción a los conceptos y la terminología de las estructuras de árbol.

6.4.2 Las partes componentes del AIG tienen las siguientes interpretaciones:

a) los vértices son los asientos. Los asientos de objeto pueden ser vértices constitutivos de hoja o vértices no constitutivos de hoja (denominados brevemente vértice-hoja y vértices-no-hoja, respectivamente), mientras que los asientos de alias son siempre vértices constitutivos de hoja. La raíz no es un asiento en sí, pero cuando sea conveniente [por ejemplo, en las definiciones de los apartados a) y b) siguientes] puede considerarse como un asiento de un objeto nulo [véase el apartado d) siguiente];

b) los arcos definen la relación entre los vértices (y, por ende, entre los asientos). Un arco del vértice A al vértice B, significa que el asiento en A es el *asiento inmediatamente superior* (*superior inmediato*) al asiento en B, e inversamente, que el asiento en B es un *asiento inmediatamente subordinado* (*subordinado inmediato*) al asiento en A. Los *asientos superiores* (*o los superiores*) de un determinado asiento son el superior inmediato de éste junto con los superiores (*de éste*) (recursivamente). Los *asientos subordinados* (*los subordinados*) de un determinado asiento son los subordinados inmediatos de éste junto con los subordinados *de éstos* (recursivamente);

c) el objeto representado por un asiento es o está estrechamente asociado con la autoridad de denominación (véase el § 8) para sus subordinados;

d) la raíz representa la existencia del más elevado nivel de autoridad de denominación para la BIG.

6.4.3 Es posible obtener una relación superior/subordinado entre objetos a partir de la relación entre asientos. Un objeto es un *objeto inmediatamente superior* (*superior inmediato*) de otro, solamente si el asiento de objeto para el primer objeto es el superior inmediato de cualquiera de los asientos para el segundo objeto. Los términos *objeto inmediatamente subordinado*, *subordinado inmediato*, *superior* y *subordinado* (aplicados a objetos) tienen significados análogos.

6.4.4 Las relaciones superior/subordinado autorizadas entre objetos se rigen por las definiciones de estructura del AIG (véase el § 9.2).

## 7 Asientos de la guía

### 7.1 *Definiciones*

a) *atributo:* la información de un tipo particular relativa a un objeto y que aparece en un asiento describiendo ese objeto en la BIG;

b) *tipo de atributo:* el componente de un atributo que indica la clase de información proporcionada por ese atributo;

c) *valor de atributo:* una instancia particular de clase de información indicada por un tipo de atributo;

d) *aserción de valor de atributo:* una proposición, que puede ser verdadera, falsa o indefinida, relativa a los valores (o quizás sólo a los valores distinguidos) de un asiento;

*Nota* - En este documento se utiliza la notación "cadena1 = cadena2" para escribir ejemplos de aserciones de valor de atributo. En esta notación, "cadena1" es una abreviatura del 'nombre' del tipo de atributo, y "cadena2" es una representación textual de un valor adecuado. Aunque los tipos de atributo en los ejemplos suelen basarse en tipos reales, como los definidos en la Recomendación X.520 (por ejemplo, "P" significa "País"; NC, "Nombre Común"), esto no es absolutamente necesario a los efectos de esta Recomendación, puesto que la guía desconoce generalmente los significados de los tipos de atributo que se están utilizando.

e) *valor distinguido:* un valor de atributo en un asiento que ha sido designado para aparecer en el nombre distinguido relativo del asiento.

### 7.2 *Estructura global*

7.2.1 Como se observa en la figura 3/X.501, un asiento consiste en un conjunto de *atributos*.

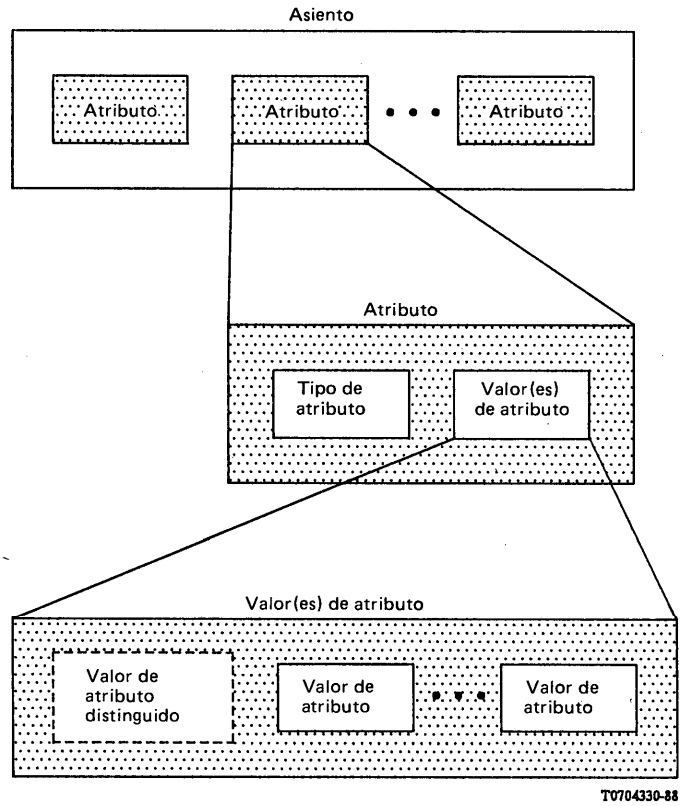



FIGURA 3/X.501

**Estructura de un asiento**

7.2.2 Cada atributo proporciona una información sobre el objeto al que corresponde el asiento, o describe una característica particular del mismo.

*Nota* - Entre los ejemplos de atributos que podrían estar presentes en un asiento figuran la información de denominación tal como el nombre personal del objeto, la información de direccionamiento, como su número de teléfono.

7.2.3 Un atributo se compone de un *tipo de atributo*, que identifica la clase de información dada por un atributo, y *el valor o los valores de atributo* correspondiente(s), que constituyen las instancias particulares de esa clase que aparecen en el asiento.

```
Attribute ::=
    SEQUENCE{
        type      Attribute Type,
        values    SET OF AttributeValue
        -- se requiere al menos un valor -- }
```

7.3 *Tipos de atributo*

7.3.1 Algunos tipos de atributo serán normalizados a nivel internacional. Otros tipos de atributo serán definidos por las autoridades administrativas nacionales y organizaciones privadas. Esto implica que cierto número de autoridades independientes serán responsables de asignar tipos de manera de lograr que cada uno sea distinto de todos los demás tipos asignados. Esto se efectúa identificando cada tipo de atributo con un identificador del objeto cuando se define el tipo (como se describe en el § 9.5):

```
AttributeType ::= OBJECT IDENTIFIER
```

7.3.2 Todos los atributos de un asiento deben pertenecer a tipos de atributo distintos.

7.3.3 Hay cierto número de tipos de atributo que la guía conoce y utiliza para sus propios fines. Estos son:

a) **Clase de objeto.** Un atributo de este tipo aparece en cada asiento, e indica la clase y superclase(s) a que pertenece el objeto.

b) **Nombre de objeto con alias.** Un atributo de este tipo aparece en cada asiento de alias, y contiene el nombre distinguido (véase el § 8.5) del objeto descrito por este asiento de alias.

Estos atributos se definen (parcialmente) en el § 9.5.4.

7.3.4 Los tipos de atributo que deben o pueden aparecer en un asiento (además de los mencionados en el § 7.3.3) se rigen por reglas que se aplican a la(s) clase(s) de objeto indicada(s).

7.4 *Valores de atributo*

7.4.1 Definir un tipo de atributo (véase el § 9.5) implica también especificar la sintaxis y, por tanto, el tipo de datos, a que deben conformarse todos los valores en esos atributos. Puede tratarse de cualquier tipo de datos:

**AttributeValue ::= ANY**

7.4.2 A lo sumo, uno de los valores de un atributo puede designarse como *valor distinguido*, en cuyo caso el valor de atributo aparece en el nombre distinguido relativo (véase el § 8.3) del asiento.

7.4.3 Una *aserción de valor de atributo* (*AVA*) es una proposición, que puede ser verdadera, falsa, indefinida, relativa a los valores (o quizás sólo a los valores distinguidos) de un asiento. Comprende un tipo de atributo y un valor de atributo.

**AttributeValueAssertion ::=**
**SEQUENCE {AttributeType, AttributeValue}**

y es:

a) indefinida, si se cumple cualquiera de las siguientes condiciones:

i) el tipo de atributo es desconocido;

ii) la sintaxis del atributo para el tipo no tiene una regla de concordancia por igualdad;

iii) el valor no se conforma al tipo de datos de la sintaxis del atributo;

*Nota* - Los elementos de los apartados ii) e iii) indican una AVA defectuosa; sin embargo, el elemento del apartado i) puede producirse como una situación local (por ejemplo, un determinado ASG no ha registrado ese tipo particular de atributo).

b) verdadera, si el asiento contiene un atributo de ese tipo, uno de cuyos valores concuerda con ese valor (si la aserción se refiere sólo a valores distinguidos, el valor con respecto al cual se determina la concordancia deberá ser el valor distinguido);

*Nota* - La concordancia de valores se determina por igualdad y comprende la norma de concordancia asociada con la sintaxis del atributo.

c) falsa, en otros casos.

## 8 Nombres

8.1 *Definiciones*

a) *alias, nombre con alias*: el nombre de un objeto, dado por el uso de uno o más asientos de alias en el AIG;

b) *desreferenciación*: sustitución del nombre de alias de un objeto por el nombre distinguido del objeto;

c) *nombre distinguido* (de un objeto): uno de los nombres del objeto, formado a partir de la secuencia de los NDR del asiento de objeto y cada uno de sus asientos superiores;

d) *nombre (en la guía)*: una construcción lingüística que permite distinguir un objeto determinado de todos los demás objetos. Un nombre tiene que ser inequívoco (o sea, designar sólo a un objeto), si bien no es necesario que sea único (o sea, el único nombre que designa inequívocamente al objeto);

e) *nombre contemplado*: una construcción que es un nombre desde el punto de vista sintáctico aunque no haya aún aparecido como un nombre válido;

f) *autoridad de denominación*: una autoridad encargada de la atribución de nombres. Todo objeto cuyo asiento se sitúa en un vértice no constitutivo de hoja en el AIG es, o está estrechamente asociado con, una autoridad de denominación;

g) *nombre distinguido relativo (NDR)*: una secuencia de aserciones de valor de atributo, todas verdaderas, relativas a los valores distinguidos de un asiento determinado.

8.2 *Nombres en general*

8.2.1 Un *nombre* (en la *guía*) es una construcción que identifica un determinado objeto en el conjunto de todos los objetos. Un nombre debe ser inequívoco, es decir, designar sólo un objeto. Sin embargo, no es necesario que el nombre sea único, o sea, que sea el único nombre que designe inequívocamente al objeto.

8.2.2 Sintácticamente, cada nombre de un objeto constituye una secuencia ordenada de nombres distinguidos relativos (véase el § 8.3).

**Name ::=**
**CHOICE {-- *sólo una posibilidad por ahora* --**
**RDSNSequence}**

**RDNSequence ::= SEQUENCE OF RelativeDistinguishedName**

**DistinguishedName ::= RDNSequence**

*Nota* - Los nombres constituidos en otras formas diferentes de las aquí descritas constituyen una posible extensión futura.

8.2.3 La secuencia nula es el nombre de la raíz del árbol.

8.2.4 Cada subsecuencia inicial del nombre de un objeto es también el nombre de un objeto. La secuencia de objetos así identificados, que comienza por la raíz y termina por el objeto que está siendo denominado, es tal que cada uno es el superior inmediato del que le sigue en la secuencia.

8.2.5 Un *nombre contemplado* es una construcción que constituye un nombre desde el punto de vista sintáctico, pero que no ha aparecido (aún) como un nombre válido.

8.3 *Nombres distinguidos relativos*

8.3.1 Cada asiento tiene un *nombre distinguido relativo (NDR)*. Un NDR consiste en una determinada secuencia de aserciones de valor de atributo, todas verdaderas, relativas a los valores distinguidos del asiento.

**RelativeDistinguishedName ::=**
**SET OF AttributeValueAssertion**

La secuencia contiene exactamente una aserción sobre cada valor distinguido en asiento.

8.3.2 Los NDR de todos aquellos asientos que tienen un mismo superior inmediato son distintos. Incumbe a la autoridad de denominación competente para el asiento considerado asegurar que esto suceda realmente, asignando de manera apropiada valores de atributo distinguidos.

*Nota* - Frecuentemente, un asiento contendrá un solo valor distinguido (y el NDR comprenderá por tanto una sola AVA); sin embargo, en ciertas circunstancias (a fin de establecer una diferenciación), pueden utilizarse valores (y por ende AVA) adicionales.

8.3.3 El NDR de un asiento se elige al crear el asiento. Una de un solo valor de cualquier tipo de atributo puede formar parte del NDR, lo que dependerá de la naturaleza de la clase de objeto designada. La atribución de NDR se considera una tarea administrativa que puede o no requerir una negociación entre las organizaciones o Administraciones que intervienen. En esta Recomendación no se describe un mecanismo de negociación semejante ni se formulan hipótesis en cuanto a su conducción. En caso necesario, el NDR podrá ser modificado, reemplazándolo completamente.

*Nota* - Los NDR están previstos para una larga duración de manera que los usuarios de la guía puedan almacenar los nombres distinguidos de objetos (por ejemplo, en la propia guía) sin preocuparse por su caducidad. Por consiguiente, deberá procederse con cautela a reemplazar los NDR.

8.4 *Nombres distinguidos*

8.4.1 El *nombre distinguido* de un objeto dado se define como la secuencia de los NDR del asiento que representa el objeto junto con las secuencias de todos sus asientos superiores (en orden descendente). Por causa de la correspondencia biunívoca entre objetos y asientos de objetos, puede considerarse que el nombre distinguido de un objeto también identifica al asiento del objeto.

*Nota 1* - Es preferible que los nombres distinguidos de los objetos que deban ser presentados a las personas resulten cómodos para éstas.

*Nota 2* - La norma ISO 7498/3 define el concepto de un nombre primitivo. Un nombre distinguido puede utilizarse como un nombre primitivo del objeto que él identifica porque: a) es inequívoco, b) es único y c) no es necesario (pero naturalmente es posible) que el usuario de la guía comprenda la semántica de su estructura interna (una secuencia de NDR).

*Nota 3* - Debido a que sólo intervienen el asiento del objeto y sus superiores, los nombres distinguidos de objetos nunca pueden comprender asientos de alias.

8.4.2 Resulta conveniente definir el "nombre distinguido" de la raíz y de un asiento de alias, aunque en ninguno de los dos casos el nombre sea también el nombre distinguido de un objeto. El nombre distinguido de la raíz se define como secuencia nula. El nombre distinguido de un asiento de alias se define como la secuencia de NDR del asiento de alias y de todos aquellos asientos superiores (en orden descendente).

8.4.3 La figura 4/X.501 presenta un ejemplo que ilustra los conceptos de NDR y nombre distinguido.

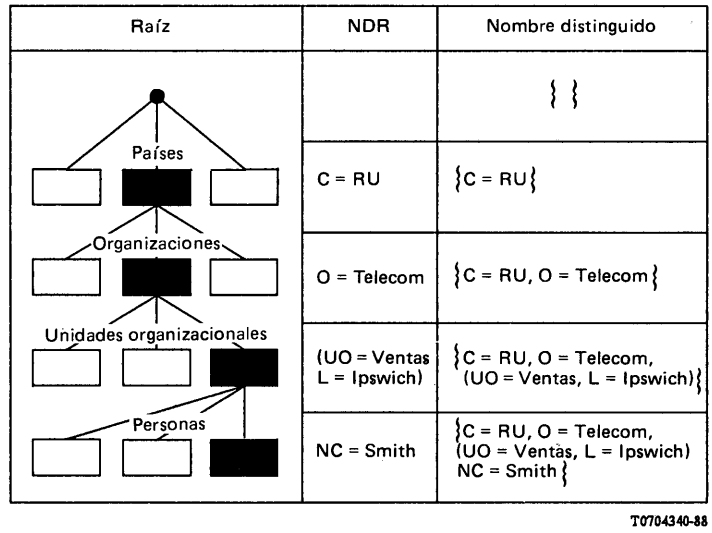



FIGURA 4/X.501

**Determinación de nombres distinguidos**

## 8.5 *Nombre de alias*

8.5.1 Un *alias*, o un *nombre de alias*, de un objeto es un nombre en el cual por lo menos uno de sus NDR es el de un asiento de alias. Los alias permiten que los asientos de objeto logren el efecto de tener múltiples superiores inmediatos. Así pues, los alias sirven de base para nombres alternativos.

8.5.2 Al igual que el nombre distinguido de un objeto expresa su relación principal con cierta jerarquía de objetos, del mismo modo un alias expresa (en el caso general) una relación alternativa con una jerarquía diferente de objetos.

8.5.3 Un objeto con un asiento en el AIG puede tener cero o más alias. Por tanto, varios asientos de alias pueden indicar el mismo asiento de objeto. Un asiento puede indicar un asiento de objeto que no es un asiento constitutivo de hoja. Solamente los asientos de objeto pueden tener alias. Por consiguiente, no se permiten alias de alias.

8.5.4 Un asiento de alias no tendrá subordinados, es decir, un asiento de alias es un asiento de hoja.

8.5.5 La guía utiliza el atributo del nombre de objeto con alias en un asiento de alias para identificar y hallar el correspondiente asiento de objeto.

## 9 Esquema de la guía

### 9.1 *Definiciones*

a) *esquema de la guía* (o plan de la guía): conjunto de reglas y restricciones relativas a la estructura del AIG, definiciones de clase de objeto, tipos de atributo y sintaxis que caracterizan la BIG;

b) *regla de estructura del AIG*: una regla que forma parte del esquema de la guía, y que relaciona una clase de objeto (el subordinado) con otra clase de objeto (el superior) y que permite que un asiento de la primera de estas dos clases quede inmediatamente subordinado a uno de la segunda, en el AIG. La regla define también el tipo o los tipos de atributo admitidos en el NDR del asiento (subordinado), y puede imponer condiciones adicionales. El esquema puede contener muchas de estas reglas.

### 9.2 *Visión de conjunto*

9.2.1 El esquema de la guía es un conjunto de definiciones y restricciones relativas a la estructura del AIG y las posibles maneras de denominar los asientos, la información que puede ser contenida en un asiento, y los atributos utilizados para representar esa información.

*Nota 1* - Por ejemplo, el esquema permite que el sistema de la guía:

- impida la creación de asientos subordinados de una clase de objeto errónea (por ejemplo, un país como subordinado de una persona);
- impida la inclusión de tipos de atributo en un asiento inapropiado para esa clase de objeto (por ejemplo, un número de serie a un asiento relativo a una persona);
- impida la adición de un valor de atributo de una sintaxis que no concuerde con la definida por el tipo de atributo (por ejemplo, una cadena imprimible a una cadena de bits).

*Nota 2* - En esta serie de Recomendaciones no se dan los mecanismos dinámicos para la gestión del esquema de la guía.

9.2.2 Formalmente, el plan de la guía comprende un conjunto de:

a) Definiciones (reglas) de *estructura del AIG* que definen los nombres distinguidos que los asientos pueden tener y las maneras en las cuales éstos pueden relacionarse entre sí a través del AIG.

b) Definiciones de *clase de objeto* que definen el conjunto de atributos obligatorios y opcionales que deben estar presentes, y pueden estar presentes, respectivamente, en un asiento de una clase dada (véase el § 6.2.3 de esta Recomendación).

c) Definiciones de *tipo de atributo* que determinan el identificador de objeto por el cual se conoce un atributo, su sintaxis y si se permite que tengan múltiples valores.

d) Definiciones de *sintaxis de atributo* que definen, para cada atributo, el tipo de datos NSA.1 y las reglas de concordancia subyacentes.

La figura 5/X.501 resume, por un lado, las relaciones entre las definiciones del plan y por otro, el AIG, asientos de guía, atributos y valores de atributo.

9.2.3 El esquema de la guía es distribuido, como la propia BIG. Cada autoridad administrativa establece la parte del esquema que aplicará para las porciones de la BIG administrados por la autoridad.

*Nota* - La distribución de información relativa al esquema, a través de ASG, gobernada por diferentes autoridades administrativas no está admitida en esta serie de Recomendaciones. Tal distribución se trata administrativamente por acuerdos bilaterales.

9.2.4 La especificación de los aspectos que intervienen en la definición de estructura del AIG, clases de objeto, tipos de atributo y sintaxis de atributo figuran en los § 9.3 a 9.6 respectivamente.

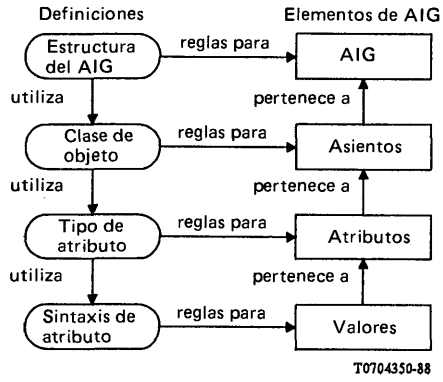



FIGURA 5/X.501

**Visión de conjunto del esquema de la guía**

9.3 *Definición de la estructura del AIG*

Una regla de estructura de AIG define las relaciones jerárquicas permitidas entre los asientos y sus NDR.

9.3.1 La definición de una regla de estructura del AIG comprende:

- identificar la clase de objeto subordinado y la clase de objeto superior;
- identificar los tipos de atributos que pueden intervenir en los NDR de asientos de subordinado; e
- información adicional (opcional).

9.3.2 La guía permite que un asiento se encuentre en la relación de subordinado inmediato a otro (que es su superior inmediato) únicamente si existe una definición de la estructura del AIG, contenida en el plan (véase el § 9.2.3), aplicable a la porción de la BIG que contendría el asiento, para el cual:

- el asiento es de la clase de objeto subordinado;
- el superior inmediato del asiento es de la clase de objeto superior;
- el (o los) tipo(s) de atributo que forma(n) el NDR del asiento se encuentra(n) entre los permitidos; y
- se cumplen todas las condiciones impuestas por el elemento del conjunto de informaciones adicionales.

*Nota 1* - Las técnicas para documentar la estructura del AIG o para representar reglas de estructura en la BIG no figuran actualmente en esta serie de Recomendaciones.

*Nota 2* - Si una regla de estructura del AIG permite subordinados o superiores inmediatos que pertenecen a una clase particular, permite también implícitamente (salvo que se especifique lo contrario) subordinados o superiores que pertenecen a cualquier clase de objeto derivada de dicha clase (véase el § 9.4).

9.3.3 La guía asegura el cumplimiento de las reglas de estructura definidas en cada asiento en el AIG. Toda tentativa de modificar el AIG de una manera que viole las reglas de estructura aplicables, fracasa.

9.3.4 Una regla de estructura del AIG en la cual una clase de objeto es el subordinado se denomina una *vinculación de nombre* para esa clase de objeto.

9.3.5 Para una clase de objeto que ha de representarse por asientos en una porción de la BIG, al menos una vinculación de nombre para esa clase de objeto debe estar contenida en la parte del esquema (o plan) aplicable. El esquema contiene las vinculaciones de nombre adicionales que sea necesario.

*Nota* - Es concebible que una clase de objeto, que ocurra en dos esquemas distintos, tenga vinculaciones de nombre distintas en cada esquema.

9.4 *Definición de clase de objeto*

9.4.1 La definición de una clase de objeto comprende:

a) opcionalmente, la asignación de un identificador de objeto para la clase de objeto;

b) la indicación de la clase de objeto de la cual ésta ha de ser una subclase;

c) el listado de los tipos de atributo *obligatorios* que un asiento de la clase de objeto debe contener además de los tipos de atributos obligatorios de todas sus superclases;

d) el listado de tipos de atributos *opcionales* que un asiento de la clase de objeto puede contener además de los atributos opcionales de todas sus superclases.

*Nota* - Una clase de objeto sin un identificador de objeto asignado está destinada al uso local, como un medio de añadir convenientemente nuevos tipos de atributo a una superclase predefinida. Esta adición permite varias posibilidades. Por ejemplo, una autoridad administrativa puede definir una clase de objeto no registrada a fin de permitir que un usuario añada al asiento cualquier atributo registrado. La autoridad administrativa puede limitar los atributos de un asiento para una clase de objeto determinada a los contenidos en una lista a nivel local. Puede también disponer que determinados atributos sean obligatorios para una determinada clase de objeto, además de los requeridos por la definición de clase de objeto registrado.

9.4.2 Hay una clase de objeto especial, de la cual toda otra clase es una subclase. Esta clase de objeto se denomina "cumbre" y se define en el § 9.4.8.1.

9.4.3 Cada asiento contendrá un atributo de tipo **clase de objeto** para identificar la clase de objeto y superclases a las cuales pertenece el asiento. La definición de este atributo figura en el § 9.5.4. El atributo tiene múltiples valores. Habrá un valor del atributo para la clase de objeto y cada una de sus superclases para las cuales se define un identificador de objeto, salvo que el valor de "cumbre" no tiene que estar presente mientras esté presente otro valor.

*Nota 1* - El requisito de que el atributo **clase de objeto** esté presente en cada asiento se refleja en la definición de **cumbre**.

*Nota 2* - Como se considera que una clase de objeto pertenece a todas sus superclases, cada miembro de la cadena de superclases hasta la "cumbre" está representado por un valor en el atributo clase de objeto (y cualquier valor de la cadena puede hacerse concordar mediante un filtro).

El atributo **clase de objeto** es proporcionado y manejado por la guía, es decir, no puede ser modificado por el usuario.

9.4.4 La guía emplea la clase de objeto definida para cada asiento en el AIG. Toda tentativa de modificar un asiento que viole la definición de clase de objeto del asiento, fracasa.

*Nota* - En particular, la guía impedirá que:

a) tipos de atributos no incluidos en la definición de clase de objeto se añadan a un asiento de esa clase de objeto;

b) se cree un asiento en el cual falten uno o más tipos de atributo que son obligatorios para la clase de objeto del asiento;

c) se suprima un tipo de atributo que es obligatorio para la clase de objeto del asiento.

9.4.5 La clase de objeto especial **alias** se define en el § 9.4.8.2. Cada asiento de alias tendrá una clase de objeto que es una subclase de esta clase.

*Nota* - La referenciación por la guía, de asientos de alias asegura que se vean raramente los valores del atributo **clase de objeto** de un asiento de alias. Se recomienda que las clases de objeto alias apropiadas se deriven del **alias** sin asignar un identificador de objeto.

9.4.6 La siguiente macro NSA.1 puede (pero no tiene necesariamente que) utilizarse para definir una clase de objeto. La producción vacía para la **SubclaseDe** está permitida solamente al definir cumbre:

```
OBJECT-CLASSMACRO ::=
BEGIN

TYPENOTATION ::= SubclassOf
                 MandatoryAttributes
                 OptionalAttributes
```

**VALUENOTATION ::=**
**value(VALUE OBJECT IDENTIFIER)**

**SubclassOf ::=**
**"SUBCLASS OF" Subclasses |**
**empty**

**Subclasses ::= Subclass | Subclass"," Subclasses**

**Subclass ::= value(OBJECT-CLASS)**

**MandatoryAttributes ::=**
**"MUST CONTAIN {"Attributes"}" | empty**

**OptionalAttributes ::=**
**"MAY CONTAIN {"Attributes"}" | empty**

**Attributes ::= AttributeTerm | AttributeTerm "," Attributes**

**AttributeTerm ::= Attribute | AttributeSet**

**Attribute ::= value(ATTRIBUTE)**

**AttributeSet ::= value(ATTRIBUTE-SET)**

**END**

La correspondencia entre las partes de la definición, tal como figura en el § 9.4.1, y las diversas piezas de la notación introducida por la macro, es la siguiente:

a) el identificador de objeto para la clase de objeto es el valor suministrado en la asignación de valor de la macro;

b) las superclases de las cuales esta clase de objeto es una subclase son las identificadas por la producción **SubclaseDe**, es decir, la que sigue a **"SUBCLASS OF"**;

c) los atributos obligatorios son los identificados por la lista de identificadores de objeto producida por la producción **AtributosObligatorios**, es decir, los que siguen a **"MUST CONTAIN"**;

d) los atributos opcionales son los identificados por la lista de identificadores de objeto producida por la producción **AtributosOpcionales**, es decir, los que siguen a **"MAY CONTAIN"**.

*Nota 1* - Los identificadores de objeto indicados en c) y d) identifican tanto atributos individuales como conjuntos de atributos (véase el § 9.4.7). La lista efectiva en ambos casos es el conjunto formado por la unión de estos conjuntos. Si un atributo aparece en el conjunto obligatorio y en el conjunto facultativo, se considerará obligatorio.

*Nota 2* - La macro se utiliza para definir clases de objeto seleccionadas en la Recomendación X.521.

Si todas las piezas de notación introducidas por la macro y descritas en b), c) y d), más arriba, están vacías, la notación resultante (**"OBJECT-CLASS"**) puede utilizarse para denotar cualquier posible clase de objeto.

9.4.7 Un *conjunto de atributos* es un conjunto de atributos identificado por un identificador de objeto. La definición de un conjunto de atributos comprende:

a) la asignación de un identificador de objeto al conjunto;

b) el listado de identificadores de objeto de los atributos y otros conjuntos de atributos cuyos miembros, juntos, forman el conjunto.

La siguiente macro NSA.1 puede (pero no tiene necesariamente que) utilizarse para definir un conjunto de atributos para uso con la macro **OBJECT CLASS**:

**ATTRIBUTE-SET-MACRO ::=**

**BEGIN**

**TYPE NOTATION ::= "CONTAINS" {"Attributes"}" | empty**

**VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)**

**Attributes ::=**
**AttributeTerm | AttributeTerm "," Attributes**

**AttributeTerm ::= Attribute | AttributeSet**

**Attribute ::= value(ATTRIBUTE)**

**AttributeSet ::= value(ATTRIBUTE-SET)**

**END**

La correspondencia entre las partes de la definición de un conjunto de atributos y la notación introducida por la macro es la siguiente:

a) el identificador de objeto asignado al conjunto de atributos es el valor suministrado en la asignación de valor de la macro;

b) el conjunto de atributos es el formado por la unión de los atributos y de los conjuntos de atributos identificados por la producción **atributos**, es decir, la que sigue a **CONTAINS**.

En caso de seleccionarse la alternativa "**vacío**", la notación resultante ("**ATTRIBUTE-SET**") puede utilizarse para denotar cualquier posible conjunto de atributos.

9.4.8 Las clases de objeto mencionadas anteriormente se definen en los § 9.4.8.1 y 9.4.8.2.

*Nota* - Estas son definiciones parciales: los identificadores de objeto se atribuyen efectivamente para estas clases de objetos en la Recomendación X.521 a fin de que haya un solo punto de atribución de estos identificadores de objeto en esta serie de Recomendaciones.

9.4.8.1 La clase de objeto **cumbre** (Top) se define como sigue:

**Top ::=**
**OBJECT-CLASS**
**MUST CONTAIN {objectClass}**

9.4.8.2 La clase de objeto **alias** se define como sigue:

**Alias ::=**
**OBJECT-CLASS**
**SUBCLASS OF top**
**MUST CONTAIN {aliasedObjectName}**

*Nota 1* - La clase de objeto **alias** no especifica tipos de atributo adecuados para el NDR de un asiento de alias. Las autoridades administrativas pueden especificar subclases de la clase "alias" que especifican tipos de atributo útiles para los NDR de asientos de alias (véase la Recomendación X.521).

*Nota 2* - Los asientos de una subclase de la clase **alias** son asientos de alias.

9.5 *Definición de tipo de atributo*

9.5.1 La definición de un tipo de atributo comprende:

a) la asignación de un identificador de objeto al tipo de atributo;

b) la indicación o la definición de la sintaxis de atributo para el tipo de atributo;

c) la indicación de si un atributo de este tipo puede tener sólo un valor o puede tener más de un valor (recurrencia).

9.5.2 La guía asegura que la sintaxis de atributo indicada se utilice para cada atributo de este tipo. La guía asegura también que los atributos de este tipo tengan un valor, y sólo uno, en los asientos, cuando los atributos de este tipo hayan sido definidos de manera que tengan un solo valor.

9.5.3 La siguiente macro NSA.1 puede (pero no tiene necesariamente que) utilizarse para definir un tipo de atributo:

**ATTRIBUTE MACRO ::=**
**BEGIN**

**TYPENOTATION ::= AttributeSyntax Multivalued | empty**

```
VALUENOTATION       ::= value (VALUE OBJECT IDENTIFIER)

AttributeSyntax     ::=
    "WITH ATTRIBUTE-SYNTAX" SyntaxChoice

Multivalued         ::= "SINGLE VALUE"
                        | "MULTIVALUE" | empty
SyntaxChoice        ::= value(ATTRIBUTE-SYNTAX)
                        Constraint | type MatchTypes
Constraint          ::= "("ConstraintAlternative")" | empty
ConstraintAlternative ::= StringConstraint | IntegerConstraint
StringConstraint    ::= "SIZE" "("SizeConstraint")"

SizeConstraint      ::= SingleValue | Range

SingleValue         ::= value(INTEGER)

Range               ::= value(INTEGER) ".." value
                        (INTEGER)

IntegerConstraint   ::= Range

MatchTypes          ::= "MATCHES FOR" Matches | empty

Matches             ::= Match Matches | Match

Match               ::= "EQUALITY" | "SUBSTRINGS" |
                        "ORDERING"

END
```

La correspondencia entre las partes de la definición, conforme al § 9.5.1, y las distintas notaciones introducidas por la macro, es la siguiente:

a) el identificador de objeto asignado al tipo de atributo es el valor suministrado en la asignación de valor de la macro;

b) la sintaxis de atributo para el tipo de atributo es la identificada por la producción de AtributeSintax. Esto indica una sintaxis de atributo definida separadamente, o explícitamente define una sintaxis de atributo dando su tipo NSA.1 y las reglas de concordancia indicadas (véase el § 9.6). Si se emplea una sintaxis de atributo identificada separadamente, puede indicarse facultativamente una restricción de tamaño para tipos cadena subyacentes o una gama de valores para un tipo entero subyacente;

c) el atributo tendrá un solo valor si la producción **MúltiplesValores** es **"SINGLE VALUE"**, y puede tener uno o más valores si es **"MULTI VALUE"** o vacío.

*Nota* - La macro se utiliza para definir tipos de atributos seleccionados en la Recomendación X.520.

En caso de seleccionarse la alternativa **"vacío"** de la notación del tipo, la notación resultante (**"ATTRIBUTE"**) puede utilizarse para denotar cualquier tipo posible de atributo.

9.5.4 Los tipos de atributo identificados en el § 7.3.3 que son conocidos y utilizados por la guía para sus propios fines, se definen como sigue:

```
ObjectClass ::= ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX objectIdentifierSyntax

AliasedObjectName ::= ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX distinguishedNameSyntax
        SINGLE VALUE
```

*Nota 1* - Estas son definiciones parciales: los identificadores de objeto son atribuidos efectivamente a estos tipos de atributo en las Recomendaciones X.520 con el fin de prever un solo punto de asignación de estos identificadores de objeto en esta serie de Recomendaciones.

*Nota 2* - Las sintaxis de atributos mencionadas en estas definiciones están definidas a su vez en el § 9.6.5.

9.6 *Definición de sintaxis de atributo*

9.6.1 La definición de una sintaxis de atributo comprende:

a) la asignación opcional de un identificador de objeto a la sintaxis de atributo;

b) la indicación del tipo de datos en NSA.1, de la sintaxis de atributo;

c) la definición de reglas apropiadas para establecer la concordancia de un valor presentado con un valor de atributo deseado, contenido en la BIG. Para una sintaxis de atributo particular podrán definirse todas, algunas o ninguna de las siguientes reglas de concordancia:

i) igualdad. Aplicable a cualquier sintaxis de atributo. El valor presentado debe ser conforme al tipo datos de la sintaxis de atributo;

ii) subcadenas. Aplicable a cualquier sintaxis de atributo con un tipo datos de cadena. El valor presentado debe ser una secuencia ("**SEQUENCE OF**"), cada uno de cuyos elementos es conforme al tipo datos;

iii) ordenación. Aplicable a cualquier sintaxis de atributo para la cual pueda definirse una regla que permitirá que se describa un valor presentado como menor, igual o mayor que un valor deseado. El valor presentado debe ser conforme al tipo datos de la sintaxis de atributo.

9.6.2 Si no se define una regla de concordancia por igualdad, la guía:

a) considera que los valores son atributos de esta sintaxis de atributo que tienen el tipo **ANY**, es decir, la guía no verifica que estos valores son conformes al tipo datos indicado para la sintaxis de atributos;

b) no tratará de establecer la concordancia de valores presentados con valores deseados de ese tipo de atributo.

*Nota* - De ahí resulta que la guía no permitirá que tal atributo se utilice en un nombre distinguido, ni autorizará a que se modifique un valor específico.

9.6.3 Si se define una regla de concordancia por igualdad, la guía:

a) considera que los valores de atributos de esta sintaxis de atributo tienen el tipo **ANY DEFINED BY** el tipo de datos indicado para la sintaxis de atributo;

b) sólo establecerá la concordancia de conformidad con las reglas de concordancia definidas para esa sintaxis de atributo;

c) sólo establecerá la concordancia de un valor presentado de un tipo datos adecuado, como se especifica en el § 9.6.1 c).

9.6.4 La siguiente macro NSA.1 puede, pero no tiene necesariamente que, utilizarse para definir sintaxis de atributo:

```
ATTRIBUTE-SYNTAX MACRO ::=
BEGIN

TYPE NOTATION ::= Syntax
                  MatchTypes | empty

VALUE NOTATION ::=
    value (VALUE OBJECT IDENTIFIER)

Syntax ::= type

MatchTypes ::= "MATCHES FOR" Matches | empty

Matches ::= Match Matches | Match

Match ::= "EQUALITY" | "SUBSTRINGS" | "ORDERING"

END
```

La correspondencia entre las partes de la definición, tal como se indica en el § 9.6.1, y las diversas piezas de la notación introducidas por la macro, es la siguiente:

a) el identificador de objeto asignado a la sintaxis de atributo es un valor suministrado en la asignación de valor de la macro;

b) el tipo datos de la sintaxis de atributo es el especificado por la producción de **sintaxis**, es decir, el que sigue al nombre de macro;

c) las reglas de concordancia definidas son por igualdad, si aparece "**EQUALITY**" en la producción **TiposdeConcordancia**, subcadenas, si aparece "**SUBSTRINGS**", y ordenación si aparece "**ORDERING**". Si la producción está vacía, no se definen reglas de concordancia.

Si se selecciona la alternativa "**vacío**" de la notacion, la notación resultante ("**ATTRIBUTE-SYNTAX**") puede utilizarse para denotar cualquier sintaxis posible de atributo.

*Nota 1* - No se proporciona soporte alguno en la macro para definir las reglas de concordancia propiamente dichas. Esto debe efectuarse en lenguaje ordinario o por otros medios.

*Nota 2* - La macro se utiliza para definir sintaxis de atributo seleccionadas en la Recomendación X.520.

9.6.5 Las sintaxis de atributo utilizadas en el § 9.5.4 se definen en los § 9.6.5.1 y 9.6.5.2.

*Nota* - Estas son definiciones parciales: los identificadores de objeto se asignan efectivamente para estas sintaxis de atributo en la Recomendación X.520, a fin de prever un solo punto de asignación de estos identificadores de objeto en la presente serie de Recomendaciones.

9.6.5.1 La **Sintaxis de Identificador de Objeto** se define como sigue:

```
ObjectIdentifierSyntax ::=
        ATTRIBUTE-SYNTAX
            OBJECT IDENTIFIER
            MATCHES FOR EQUALITY
```

La regla de concordancia por igualdad es inherente a la definición de identificador de objeto en NSA.1.

9.6.5.2 La **Sintaxis de Nombre Distinguido** se define como sigue:

```
DistinguishedNameSyntax
        ATTRIBUTE-SYNTAX
            DistinguishedName
            MATCHES FOR EQUALITY
```

Un valor de nombre distinguido presentado es igual a un valor de nombre distinguido deseado únicamente si se cumplen todas las condiciones siguientes:

a) el número de NDR en cada uno es el mismo;

b) los NDR correspondientes tienen el mismo número de AVA;

c) las AVA correspondientes (es decir, las de tipos de atributo idénticos) tienen valores de atributo que concuerdan por igualdad (en esta concordancia, los valores de atributo desempeñan los mismos papeles, es decir, de valor presentado o deseado, que desempeña el nombre distinguido que los contiene, en la concordancia global).

SECCION 3 - *Modelo de seguridad*

## 10 Seguridad

10.1 La guía existe en un entorno en el que diversas autoridades proporcionan acceso a su parte de la BIG. Este acceso será conforme a la política de seguridad (véase la Recomendación X.509) del dominio de seguridad en el cual existe esa parte de la BIG.

10.2 Se tratan aquí dos componentes específicos de la política de seguridad:

a) la definición de una política de autorización;

b) la definición de una política de autenticación.

10.3 La definición de autorización en el contexto de la guía comprende los métodos para:

a) especificar derechos de acceso;

b) hacer cumplir los derechos de acceso (control de acceso);

c) mantener los derechos de acceso.

10.4 La definición de autenticación en el contexto de la guía comprende los métodos para verificar:

a) la identidad de los ASG y usuarios de la guía;

b) la identidad del origen de la información recibida en un punto de acceso.

La integridad de la información recibida es un asunto local y se ajustará a la política de seguridad en vigor.

10.5 Esta Recomendación no define una política de seguridad.

10.6 En el anexo F se dan orientaciones para especificar los derechos de acceso.

10.7 La Recomendación X.509 define los procedimientos de autenticación. El PAG y el PSG pueden proporcionar una autenticación fuerte del iniciador mediante la firma de la petición, y de la integridad de datos de la petición mediante la firma de la petición, autenticación fuerte del respondedor y de la integridad de datos del resultado mediante la firma del resultado. El PAG puede proporcionar una autenticación simple entre un AUG y un ASG. El PSG puede proporcionar una autenticación simple entre dos ASG.

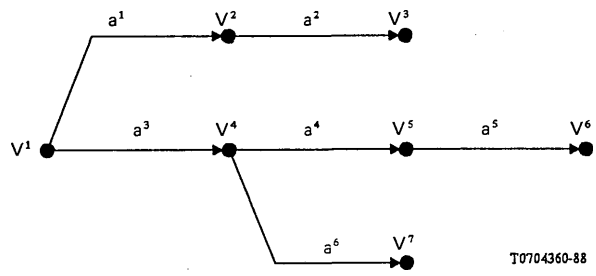
10.8 Las autoridades administrativas de aplicaciones que utilizan la guía pueden seguir su propia política de seguridad. La guía puede soportar aplicaciones manteniendo la información de autenticación (por ejemplo, nombres distinguidos, contraseñas, certificados) relativos a entidades de comunicación. Esto se describe más detalladamente en la Recomendación X.509.

## ANEXO A

(a la Recomendación X.501)

**Las matemáticas de los árboles**

Este anexo no forma parte de la Recomendación.




Un árbol es un conjunto de puntos, llamados *vértices*, y un conjunto de líneas dirigidas llamadas *arcos*; cada arco a conduce de un vértice V a un vértice V′. Por ejemplo, el árbol de la figura tiene 7 vértices, señalados con $V^1$ a $V^7$, y 6 arcos, señalados $a^1$ a $a^6$.

Los dos vértices V y V′ se denominan vértice *inicial* y vértice *final*, respectivamente, de un arco a de V a V′. Por ejemplo, $V^2$ y $V^3$ son los vértices inicial y final, respectivamente del arco $a^2$. Varios arcos diferentes pueden partir del mismo vértice inicial, pero no tendrán el mismo vértice final. Por ejemplo, el arco $a^1$ y $a^3$ tiene el mismo vértice inicial $V^1$, pero no hay dos arcos en la figura que tengan el mismo vértice final.

El vértice que no es el vértice final de ningún arco suele llamarse vértice *raíz*, o de manera aún más informal, la "raíz" del árbol. Por ejemplo, en la figura, $V^1$ es la raíz.

Un vértice que no es el vértice inicial de ningún arco suele llamarse informalmente vértice *hoja*, o de manera aún más informal, una "hoja" de la gráfica del árbol. Por ejemplo, los vértices $V^3$, $V^6$ y $V^7$ son hojas.

Un *trayecto orientado* desde un vértice V a un vértice V' es un conjunto de arcos $(a^1,a^2, ..., a^n)$ $(n \geq 1)$ tal que V es el vértice inicial del arco $a^1$ y V' es el vértice final del arco $a^n$, y el vértice final del arco $a^k$ es también el vértice inicial del arco $a^{k+1}$ siendo $1 \leq k < n$. Por ejemplo, el trayecto orientado del vértice $V^1$ al vértice $V^6$ es el conjunto de arcos $(a^3,a^4,a^5)$. Se entenderá que el término "trayecto" designa a un trayecto orientado desde la raíz hacia un vértice.

## ANEXO B

(a la Recomendación X.501)

### Utilización del identificador de objeto

El presente anexo forma parte de la Recomendación.

Este anexo documenta la parte superior del subárbol identificador de objeto en que residen todos los identificadores de objetos asignados en esta serie de Recomendaciones. Esto se efectúa proporcionando un módulo NSA.1 llamado **"UsefulDefinitions"** en el cual se asignan nombres a todos los nodos no constitutivos de hojas del subárbol.

```
UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)  usefulDefinitions(0)}
DEFINITIONS ::=
BEGIN

EXPORTS
      module, serviceElement, applicationContext, attribute Type,
      attributeSyntax, objectClass, algorithm, abstractSyntax, attributeSet,

      usefulDefinitions, informationFramework, directoryAbstractService,
      directoryObjectIdentifiers, algorithmObjectIdentifiers, distributedOperations,
      protocolObjectIdentifiers, selectedAttributeTypes, selectedObjectClasses,
      authenticationFramework, upperBounds,dap,dsp.

      id-ac,id-ase,id-as, id-ot, id-pt;

ds          OBJECT IDENTIFIER ::=      {joint-iso-ccitt ds(5)}

-- categorías de objeto de información --

module              OBJECT IDENTIFIER ::= {ds 1}
serviceElement      OBJECT IDENTIFIER ::= {ds 2}
applicationContext  OBJECT IDENTIFIER ::= {ds 3}
attributeType       OBJECT IDENTIFIER ::= {ds 4}
attributeSyntax     OBJECT IDENTIFIER ::= {ds 5}
objectClass         OBJECT IDENTIFIER ::= {ds 6}
attributeSet        OBJECT IDENTIFIER ::= {ds 7}
algorithm           OBJECT IDENTIFIER ::= {ds 8}
abstractSyntax      OBJECT IDENTIFIER ::= {ds 9}
object              OBJECT IDENTIFIER ::= {ds 10}
port                OBJECT IDENTIFIER ::= {ds 11}
```

```
-- módulos --

usefulDefinitions                   OBJECT IDENTIFIER ::= {módulo 0}
informationFramework                OBJECT IDENTIFIER ::= {módulo 1}
directoryAbstractService            OBJECT IDENTIFIER ::= {módulo 2}
distributedOperations               OBJECT IDENTIFIER ::= {módulo 3}
protocolObjectIdentifier            OBJECT IDENTIFIER ::= {módulo 4}
selectedAttributeTypes              OBJECT IDENTIFIER ::= {módulo 5}
selectedObjectClasses               OBJECT IDENTIFIER ::= {módulo 6}
authenticationFramework             OBJECT IDENTIFIER ::= {módulo 7}
algorithmObjectIdentifiers          OBJECT IDENTIFIER ::= {módulo 8}
directoryObjectIdentifiers          OBJECT IDENTIFIER ::= {módulo 9}
upperBounds                         OBJECT IDENTIFIER ::= {módulo 10}
dap                                 OBJECT IDENTIFIER ::= {módulo 11}
dsp                                 OBJECT IDENTIFIER ::= {módulo 12}
distributedDirectoryObjectIdentifier OBJECT IDENTIFIER ::= {módulo 13}

-- sinónimos --

id-ac       OBJECT IDENTIFIER ::= applicationContex
id-ase      OBJECT IDENTIFIER ::= serviceElement
id-as       OBJECT IDENTIFIER ::= abstractSyntax
id-ot       OBJECT IDENTIFIER ::= object
id-pt       OBJECT IDENTIFIER ::= port

END
```

## ANEXO C

(a la Recomendación X.501)

### Marco de información (information framework) en NSA.1

Este anexo forma parte de la Recomendación.

En este anexo se presenta un resumen de todas las definiciones de tipo, valor y macro, en NSA.1, contenidas en esta Recomendación. Las definiciones forman el módulo NSA.1 "InformationFramework".

```
InformationFramework {joint-iso-ccitt ds(5) modules(1)
                      informationFramework(1)}
DEFINITIONS ::=
BEGIN

EXPORTS
    Attribute, AttributeType, AttributeValue, AttributeValueAssertion,
    DistinguishedName, Name, RelativeDistinguishedName,
    OBJECT-CLASS, ATTRIBUTE, ATTRIBUTE-SET, ATTRIBUTE-SYNTAX,
    Top, Alias,
    ObjectClass, AliasedObjectName,
    ObjectIdentifierSyntax, DistinguishedNameSyntax;

IMPORTS
    selectedAttributeTypes, selectedObjectClasses
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)
            usefulDefinitions(0)}
    top
        FROM SelectedObjectClasses selectedObjectClasses
    objectIdentifierSyntax, distinguishedNameSyntax, objectClass, aliasedObjectName
        FROM SelectedAttributeType selectedAttributeTypes;
```

```
-- tipos de datos de atributo --

Attribute                   ::=   SEQUENCE{
                                      type            AttributeType,
                                      values SET OF AttributeValue
                                      -- at least one value is required --}

AttributeType               ::=   OBJECT IDENTIFIER

AttributeValue              ::=   ANY

AttributeValueAssertion     ::=   SEQUENCE {AttributeType, AttributeValue}

-- tipos de datos de denominación --

Name                        ::=   CHOICE {-- only one possibility for now --
                                      RDNSequence}

RDNSequence                 ::=   SEQUENCE OF
                                      RelativeDistinguishedName

DistinguishedName           ::=   RDNSequence

RelativeDistinguishedName   ::=   SET OF AttributeValueAssertion

-- macros --

OBJECT-CLASS MACRO          ::=
BEGIN
    TYPENOTATION            ::=   SubclassOf MandatoryAttributes
                                      OptionalAttributes
    VALUENOTATION           ::=   value (VALUE OBJECT IDENTIFIER)

    SubclassOf              ::=   "SUBCLASS OF" Subclasses | empty
    Subclasses              ::=   Subclass | Subclass "," Subclasses
    Subclass                ::=   value (OBJECT-CLASS)
    MandatoryAttributes     ::=   "MUST CONTAIN {"Attributes"}" | empty
    OptionalAttributes      ::=   "MAY CONTAIN {"Attributes"}" | empty

    Attributes              ::=   AttributeTerm | AttributeTerm "," Attributes
    AttributeTerm           ::=   Attribute | AttributeSet
    Attribute               ::=   value (ATTRIBUTE)
    AttributeSet            ::=   value (ATTRIBUTE-SET)
END

ATRIBUTE-SET MACRO          ::=
    BEGIN
    TYPENOTATION            ::=   "CONTAINS {Attributes"}" | empty
    VALUENOTATION           ::=   value(VALUEOBJECTIDENTIFIER)
    Attributes              ::=   AttributeTerm | AttributeTerm "," Attributes
    AttributeTerm           ::=   Attribute | AttributeSet
    Attribute               ::=   value(ATTRIBUTE)
    AttributeSet            ::=   value(ATTRIBUTE-SET)
    END

ATTRIBUTE MACRO             ::=
BEGIN
TYPENOTATION                ::=   AttributeSyntax Multivalued | empty
VALUENOTATION               ::=   value(VALUE OBJECT IDENTIFIER)
AttributeSyntax             ::=   "WITH ATTRIBUTE-SYNTAX" SyntaxChoice
Multivalued                 ::=   "SINGLE VALUE" | "MULTIVALUE" | empty
SyntaxChoice                ::=   value(ATTRIBUTE-SYNTAX)
                                  Constraint | type Match Types
```

```
Constraint              ::=  "("ConstraintAlternative")" | empty
ConstraintAlternative   ::=  StringConstraint | IntegerConstraint
StringConstraint        ::=  "SIZE" "("SizeConstraint")"
SizeConstraint          ::=  SingleValue | Range
SingleValue             ::=  value (INTEGER)
Range                   ::=  value (INTEGER) ".." value(INTEGER)
IntegerConstraint       ::=  Range
MatchTypes              ::=  "MATCHES FOR" Matches | empty
Matches                 ::=  Match Matches | Match
Match                   ::=  "EQUALITY" | "SUBSTRINGS" | "ORDERING"
END

ATTRIBUTE-SYNTAX MACRO  ::=
BEGIN
    TYPENOTATION        ::=  Syntax MatchTypes | empty
    VALUENOTATION       ::=  value (VALUE OBJECT IDENTIFIER)
    Syntax              ::=  type
    MatchTypes          ::=  "MATCHES FOR" Matches | empty
    Matches             ::=  Match Matches | Match
    Match               ::=  "EQUALITY" | "SUBSTRINGS" | "ORDERING"
END

-- clases de objeto --

Top         ::=  OBJECT-CLASS
                     MUST CONTAIN{objectClass}
Alias       ::=  OBJECT-CLASS
                     SUBCLASS OF top
                     MUST CONTAIN{aliasedObjectName}

-- tipos de atributo --

ObjectClass         ::=  ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX objectIdentifierSyntax
AliasedObjectName   ::=  ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX distinguishedNameSyntax
                             SINGLE VALUE

-- sintaxis de atributo --

ObjectIdentifierSyntax ::=
    ATRIBUTE-SYNTAX
    OBJECT IDENTIFIER
    MATCHES FOR EQUALITY
DistinguishedNameSyntax ::=
    ATTRIBUTE-SYNTAX
    DistinguishedName
    MATCHES FOR EQUALITY
END
```

## ANEXO D

(a la Recomendación X.501)

### Indice alfabético de definiciones

El presente anexo no forma parte de la Recomendación.

En este anexo se indican en orden alfabético todos los términos definidos en esta Recomendación, con una referencia a la cláusula en que aparecen definidos.

ANEXO E

(a la Recomendación X.501)

## Criterios de diseño de nombres

Este anexo no forma parte de la Recomendación.

El marco de información es muy general y admite una diversidad arbitraria de asientos y atributos en el AIG. Dado que, según lo aquí definido, los nombres guardan estrecha relación con los trayectos a través del AIG, es posible esa diversidad arbitraria de nombres. En esta sección se sugieren los criterios que deben considerarse para el diseño de nombres. Se han utilizado criterios adecuados para el diseño de las formas de nombre recomendadas que figuran más adelante en el presente documento. Se sugiere que los criterios también se utilicen, cuando proceda, para el diseño de nombres de objetos a los que no se aplican las formas de nombre recomendadas.

Por el momento se sigue un solo criterio, el de la comodidad para el usuario.

*Nota* - No todos los nombres tienen que ser cómodos para el usuario.

### E.1 *Comodidad para el usuario*

Los nombres con los que tienen que trabajar directamente las personas deben ser cómodos para el usuario. Un nombre cómodo para el usuario es el que toma en consideración el punto de vista del usuario humano y no el del computador. Es un nombre fácil de deducir, recordar y comprender por las personas, y no uno que sea fácil de interpretar por los computadores.

El objetivo de la "comodidad para el usuario" puede exponerse con un poco más de precisión en base a los dos principios siguientes:

- Un individuo debe por lo general estar en condiciones de encontrar el nombre "cómodo para el usuario" de un objeto partiendo de la información que posee naturalmente acerca del objeto. Por ejemplo, debe poder "adivinar" el nombre de una persona dedicada a algún negocio sólo con la información que adquiera naturalmente sobre esa persona por una asociación normal en materia de negocios.

- Cuando un nombre de objeto se especifica de manera ambigua, la guía debe reconocer esa situación en vez de llegar a la conclusión de que el nombre identifica a un objeto determinado. Por ejemplo, cuando dos personas tienen el mismo apellido, el apellido solo deberá considerarse una identificación insuficiente para cada una de esas dos personas.

Del objetivo primordial de la comodidad para el usuario se desprenden los siguientes objetivos de segundo orden:

a) Los nombres no deben eliminar artificialmente las ambigüedades naturales. Por ejemplo, si dos personas poseen el mismo apellido "Rodríguez", no se pedirá a ninguno de los dos que respondan a "JRodríguez" o "Rodríguez2". En lugar de ello, el convenio de denominación deberá proporcionar un medio "cómodo para el usuario" de distinguir las entidades. Por ejemplo, puede requerirse el primer nombre y la inicial del segundo nombre, además del apellido.

b) Es preciso que los nombres admitan abreviaturas comunes y variantes comunes de su ortografía. Por ejemplo, si una persona está empleada por la Conway Steel Corporation y el nombre de la empresa acompaña al nombre de esa persona, cualquiera de los nombres "Conway Steel Corporation", "Conway Steel Corp.", "Conway Steel" y "CSC" deberían bastar para identificar a la organización de que se trata.

c) En ciertos casos se pueden utilizar nombres de alias: para orientar la búsqueda de un asiento en particular, para una mayor comodidad para el usuario, o para reducir el ámbito de la búsqueda. El siguiente ejemplo muestra el uso de nombres de alias para tal efecto: como se ve en la figura E-1/X.501, la sucursal de Osaka también se puede identificar con el nombre {P = Japón, L = Osaka, O = ABC, UO = Agencia de Osaka}.

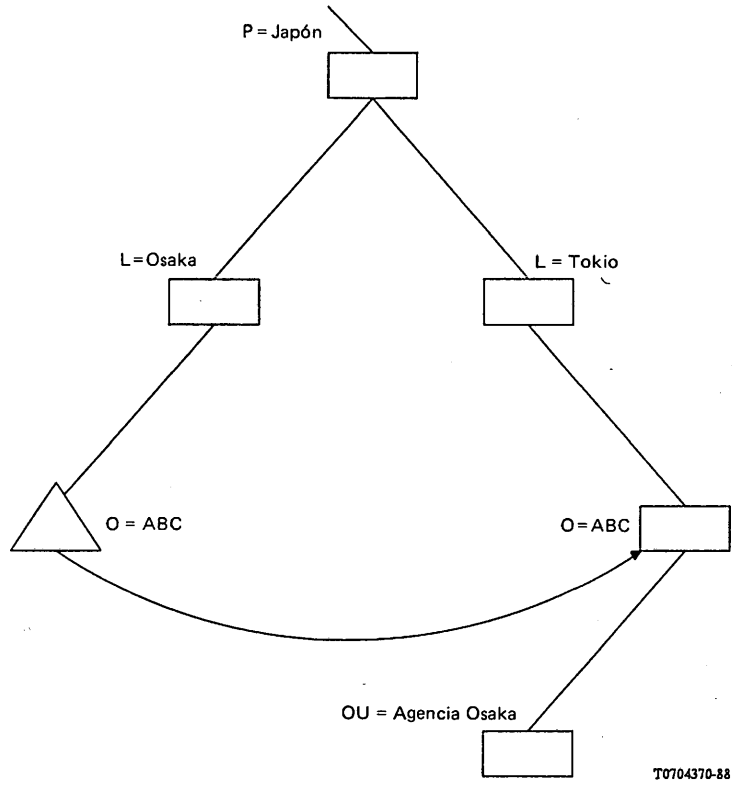



FIGURA E-1/X.501

**Ejemplo de empleo de alias**

d) En el caso de los nombres compuestos, tanto el número de los componentes obligatorios como el de los facultativos deberá ser relativamente pequeño, con lo cual serán más fáciles de recordar.

e) Si los nombres tienen varios componentes, por lo general el orden preciso en que aparecen dichos componentes deberá ser intrascendente.

f) Los nombres cómodos para el usuario no deben incluir direcciones de computador.

## ANEXO F

(a la Recomendación X.501)

**Control de acceso**

Este anexo no forma parte de la Recomendación.

### F.1 *Introducción*

A los usuarios de la guía se concede acceso a la información en la BIG sobre la base de sus derechos de control de acceso de acuerdo con la política de control de acceso en vigor, que protege dicha información.

El control de acceso es un asunto local en esta serie de Recomendaciones. Sin embargo, se reconoce que las implementaciones tendrán que introducir medios para controlar el acceso y que es probable que las futuras versiones de esta serie de Recomendaciones definan medios normalizados para crear, mantener y aplicar información de control de acceso. En este anexo se describen los principios que sirven de fundamento al control de acceso y se exponen dos posibles métodos para el control de acceso.

F.2 *Principios*

Los dos principios que guiarán el establecimiento de procedimientos para menejar el control de acceso son:

a) debe haber medios para proteger la información de la guía contra una detección, examen y modificación no autorizados, incluida la protección del AIG contra una modificación no autorizada;

b) la información requerida para determinar los derechos de un usuario a efectuar una operación dada deben estar a la disposición de los ASG que participan en la realización de la operación a fin de evitar ulteriores operaciones a distancia con la única finalidad de determinar estos derechos.

F.3 *Niveles de protección e items protegidos*

Actualmente se identifican los siguientes niveles de protección:

a) protección de un subárbol completo del AIG;

b) protección de un asiento en particular;

c) protección de un atributo completo dentro de un asiento;

d) protección de instancias seleccionadas de valores de atributo.

F.4 *Categorías de acceso*

Se prevé la necesidad de cinco categorías de acceso, por lo menos. Si no se concede acceso a un ítem protegido en cualquier categoría, la guía responde, en la medida posible, como si su ítem protegido no existiese en absoluto.

Las categorías de acceso se muestran en el cuadro F-1/X.501. La columna de ítems denota si el ítem que puede protegerse así es un asiento (E), un atributo (A) o ambos (EA).

CUADRO F.1/X.501

**Categorías de acceso**

| Categoría | Items | Descripción |
|---|---|---|
| detectar | A | Permite detectar el elemento protegido. |
| comparar | A | Permite comparar un valor presentado con el ítem protegido. |
| leer | A | Permite leer el ítem protegido. |
| modificar | A | Permite actualizar el ítem protegido. |
| añadir/suprimir | EA | Permite la creación y supresión de nuevos componentes (atributos o valores de atributo) dentro del ítem protegido. |
| denominar | E | Permite la modificación del nombre distinguido relativo de, y la creación y supresión de, asientos que están inmediatamente subordinados al asiento protegido. |

F.5 *Determinación de derechos de acceso*

Un plan para manejar el control de acceso asocia a cada ítem protegido, explícita o implícitamente, una lista de derechos de acceso. Cada ítem en esa lista aparea un conjunto de usuarios con un conjunto de categorías de acceso.

A partir de la información suministrada con la petición, ya sea la identidad y credenciales autenticadas del usuario suministradas en BIND, o la información transportada en el argumento de la operación, será posible determinar si un usuario está en uno (o más) de los conjuntos indicados.

Existen por lo menos dos posibilidades:

a) Los conjuntos se describen en base a los nombres distinguidos de los usuarios identificados por ellos, sea el nombre distinguido del usuario o el nombre distinguido de un superior con una bandera que especifica que está incluido todo el subárbol.

b) Los conjuntos indican solamente una capacidad e incluyen implícitamente a todos los usuarios que tienen dicha capacidad. Este esquema requiere que esa capacidad de usuario esté disponible localmente o bien que sea transportada en el argumento de BIND o de la operación. Esto último puede requerir una ampliación de los protocolos actualmente definidos.

**Recomendación X.509**

## LA GUIA - MARCO DE AUTENTICACION [1)]

*(Melbourne, 1988)*

INDICE



---

1) La Recomendación X.509 y la norma ISO 9594-8, Information Processing Systems - Open Systems Interconnection - The Directory-Authentication Framework (Sistemas de procesamiento de información - Interconexión de sistemas abiertos - La Guía - Marco de autenticación) se redactaron en estrecha colaboración y están técnicamente alineadas.

## 0 Introducción

0.1 Esta Recomendación, junto con las otras de la serie, ha sido elaborada para facilitar la interconexión de los sistemas de procedimiento de información para suministrar servicios de guía. El conjunto de tales sistemas, junto con la información de guía que contienen, puede ser visto como un todo integrado, llamado la *guía*. La información contenida por la guía, conocida colectivamente por la base de información de la guía (BIG), se usa típicamente para facilitar la comunicación entre, con o sobre objetos tales como entidades de aplicación de ISA, personas, terminales y listas de distribución.

0.2 La guía desempeña un papel importante en la Interconexión de Sistemas Abiertos cuyo objetivo es el permitir, con un mínimo de concordancia técnica fuera de las propias normas de interconexión, la interconexión de sistemas de procesamiento de información:

- de diferentes fabricantes;
- sometidos a gestiones diferentes;
- de diferentes grados de complejidad; y
- de diferentes fechas de construcción.

0.3 Muchas aplicaciones tienen exigencias de seguridad para la protección contra las amenazas a la comunicación de información. Algunas amenazas comúnmente conocidas, junto con los servicios de seguridad y los mecanismos que se pueden utilizar contra ellos, se describen brevemente en el anexo A. Virtualmente, todos los servicios de seguridad dependen de que las identidades de las partes comunicantes sean fiablemente conocidas, es decir, de la autenticación.

0.4 Esta Recomendación define un marco para el suministro de servicios de autenticación por la guía a sus usuarios. Estos usuarios incluyen la propia guía, así como otras aplicaciones y servicios. Incumbe a la guía la satisfacción de sus necesidades de autenticación y de otros servicios de seguridad, porque es el lugar natural del cual las partes comunicantes pueden obtener la información de autenticación de cada una de las demás: el conocimiento que es la base de la autenticación. La guía es el lugar natural porque ella contiene otras informaciones que se requieren para la comunicación y se obtienen con anterioridad al inicio de la comunicación. La obtención de la información de autenticación de un copartícipe potencial en la comunicación, desde la guía, es, con este enfoque, similar a la obtención de una dirección. Debido al vasto alcance de la guía para los fines de comunicación, se espera que este marco de autenticación será ampliamente usado por una gama de aplicaciones.

## 1 Alcance y campo de aplicación

1.1 Esta Recomendación:

- especifica la forma de la información de autenticación contenida por la guía;
- describe cómo puede obtenerse la información de autenticación a partir de la guía;
- enuncia los supuestos formulados en cuanto a la formación y al emplazamiento de esa información de autenticación en la guía;
- define tres modos en los cuales las aplicaciones pueden usar esa información de autenticación para realizar la autenticación, y describe cómo otros servicios de seguridad pueden ser soportados por autenticación.

1.2 Esta Recomendación describe dos niveles de autenticación: autenticación simple, mediante el uso de una contraseña como verificación de una identidad alegada, y autenticación fuerte, que implica credenciales formadas usando técnicas criptográficas. Si bien la autenticación simple ofrece cierta protección limitada contra el acceso no autorizado, sólo la autenticación fuerte debe servir de base para ofrecer servicios seguros. No se pretende con ello establecer un marco general para la autenticación; no obstante, puede ser de uso general para aplicaciones en que estas técnicas se consideran adecuadas.

1.3 La autenticación (y otros servicios de seguridad) sólo puede suministrarse dentro del contexto de una política de seguridad definida para una aplicación particular. Incumbe a los usuarios de una aplicación definir su propia política de seguridad, la cual puede verse limitada por los servicios proporcionados según una norma.

1.4 Incumbe a las normas definir las aplicaciones que usan el marco de autenticación para especificar los intercambios de protocolo que necesitan ser realizados para lograr la autenticación basada en la información de autenticación de la guía. El protocolo usado por las aplicaciones para obtener la información de autenticación de la guía es el protocolo de acceso a la guía (PAG), especificado en la Recomendación X.519.

1.5 El método de autenticación fuerte especificado en esta Recomendación se basa en los criptosistemas de claves públicas. Es una gran ventaja de esos sistemas el que los certificados de usuario puedan estar contenidos en la guía como atributos, y ser comunicados libremente dentro del sistema de la guía y obtenidos por los usuarios de la guía del mismo modo que otra información de guía. Se supone que los certificados de usuario están formados por medios "fuera-de-línea", y que son introducidos en la guía por su creador. La generación certificados de usuario la efectúa cierta autoridad de certificación "fuera-de-línea" que está completamente separada de los ASG en la guía. En particular, no se imponen requisitos especiales a los suministradores de la guía para almacenar o comunicar certificados de usuario en una forma segura.

En el anexo B se presenta una breve introducción a la criptografía de claves públicas.

1.6 En general, el marco de autenticación no depende del uso de un determinado algoritmo criptográfico, siempre que tenga las propiedades descritas en el § 6.1. Es probable, en la práctica, que se use cierto número de algoritmos diferentes. Sin embargo, dos usuarios que quieran autenticar tienen que soportar el mismo algoritmo criptográfico para que la autenticación se realice correctamente. Así, dentro del contexto de un conjunto de aplicaciones conexas, la elección de un algoritmo único servirá para maximizar la comunidad de usuarios capaces de autenticar y comunicar de manera segura. En el anexo C se presenta un ejemplo de un algoritmo criptográfico de claves públicas.

1.7 Análogamente, dos usuarios que deseen autenticar tienen que soportar la misma función hash (véase el § 3.3 f)) (usada en la formación de credenciales y testigos de autenticación). Aquí también, en principio, un número de funciones hash alternativas pudieran ser usadas, a expensas de reducir las comunidades de usuarios capaces de autenticar. En el anexo D se presenta una breve introducción a las funciones hash, así como un ejemplo de función hash.

## 2 Referencias

2.1 ISO 7498-2 Information Processing Systems - Open Systems Interconnection - Security Architecture (Sistemas de procesamiento de información - interconexión de sistemas abiertos - arquitectura de seguridad).

## 3 Definiciones

3.1 Esta Recomendación emplea los siguientes términos generales relacionados con la seguridad definidos en la Parte 2 del Modelo de Referencia de ISA para Seguridad:

a) *asimétrico* (cifrado);

b) *intercambio de autenticaciones;*

c) *información de autenticación;*

d) *confidencialidad;*

e) *credenciales;*

f) *criptografía;*

g) *autenticación del origen de datos;*

h) *descifrado;*

i) *cifrado;*

j) *clave;*

k) *contraseña;*

l) *autenticación de entidad par;*

m) *simétrico* (cifrado).

3.2 Los siguientes términos usados en esta Recomendación se definen en la Recomendación X.501:

a) *atributo;*

b) *base de información de la guía;*

c) *árbol de información de la guía;*

d) *nombre distinguido;*

e) *asiento;*

f) *objeto;*

g) *raíz.*

3.3 Los siguientes términos específicos se definen y usan en esta Recomendación:

a) *testigo de autenticación (testigo):* la información transportada durante un intercambio de autenticación fuerte y que puede usarse para autenticar a quien la envió;

b) *certificado de usuario (certificado):* la clave pública de un usuario, junto con alguna otra información, hecha infalsificable por cifrado con la clave secreta de la autoridad de certificación que la emitió;

c) *autoridad de certificación:* una autoridad a la cual uno o más usuarios han confiado la creación y asignación de certificados. Opcionalmente, la autoridad de certificación puede crear las claves de los usuarios;

d) *trayecto de certificación:* una secuencia ordenada de certificados de objetos en el AIG la cual, junto con la clave pública del objeto inicial en el trayecto, puede ser procesada para obtener la del objeto final en el trayecto;

e) *sistema criptográfico, criptosistema:* una colección de transformaciones de texto ordinario en texto cifrado y viceversa, seleccionándose por claves la transformación o transformaciones a ser usadas. Las transformaciones se definen normalmente por un algoritmo matemático;

f) *función hash:* una función (matemática) que hace corresponder valores de un dominio vasto (que puede ser muy vasto) con una gama menor. Una función hash 'buena' es aquella que cuando se aplica a un conjunto (grande) de valores en el dominio, los resultados se distribuyen uniformemente (y aparentemente al azar) en toda la gama;

g) *función unidireccional:* una función (matemática) que es fácil de computar, pero que, para un valor general "y" en la gama, es computacionalmente difícil hallar, en el dominio, un valor x tal que f(x)=y. Puede haber unos pocos valores "y" para los cuales hallar x no sea computacionalmente difícil;

h) *clave pública:* (en un criptosistema de claves públicas) la clave, de un par de claves de un usuario, que se conoce públicamente;

i) *clave privada (clave secreta - desaconsejada):* (en un criptosistema de claves públicas) la clave, de un par de claves de un usuario, que es conocida solamente por ese usuario;

j) *autenticación simple:* autenticación por medio de arreglos de contraseñas simples;

k) *política de seguridad:* el conjunto de reglas establecidas por la autoridad de seguridad que rigen la utilización y prestación de servicios y facilidades de seguridad;

l) *autenticación fuerte:* autenticación por medio de credenciales derivadas criptográficamente;

m) *fiduciario:* generalmente, se puede decir que una entidad acepta como "fiduciaria" a una segunda entidad cuando aquélla (la primera entidad) confía en que la segunda entidad se comportará exactamente como ella lo espera. Esta relación fiduciaria puede que sea aplicable solamente para alguna función específica. El papel principal de la confianza en el marco de la autenticación es el de describir la relación entre la entidad autenticadora y una autoridad de certificación; una entidad autenticadora tiene que estar segura de que puede confiar en que la autoridad de certificación creará solamente certificados válidos y fiables.

n) *número secuencial de certificado:* valor entero, único en la AC expedidora, que va asociado inequívocamente a un certificado expedido por dicha AC.

## 4 Notación y abreviaturas

4.1 La notación usada en esta Recomendación se define en el cuadro 1/X.509.

*Nota* - Cuando se introducen las notaciones, los símbolos X, $X_1$, $X_2$, etc., aparecen en lugar de los nombres de los usuarios, mientras que el símbolo I aparece en lugar de una información arbitraria.

4.2 En esta Recomendación se usan las siguientes abreviaturas:

AC Autoridad de certificación

AIG Arbol de información de la guía

BIG Base de información de la guía

CSCP Criptosistema de claves públicas

CUADRO 1/X.509

**Notación**

| NOTACION | SIGNIFICADO |
|---|---|
| Xp | clave pública de un usuario X |
| Xs | clave secreta de X |
| Xp[I] | cifrado de alguna información, I, mediante la clave pública de X |
| Xs[I] | cifrado de I mediante la clave secreta de X |
| X[I] | la firma de I por el usuario de X. Consiste en I con un sumario encifrado añadido |
| CA(X) | una autoridad de certificación del usuario X |
| $CA^{n}(X)$ | (donde n > 1): AC(AC(...n veces ...(X))) |
| $X_1\langle\langle X_2\rangle\rangle$ | el certificado de usuario $X_2$ emitido por la autoridad de certificación $X_1$ |
| $X_1\langle\langle X_2\rangle\rangle X_2\langle\langle X_3\rangle\rangle$ | una cadena de certificados (puede tener una longitud arbitraria), donde cada ítem es el certificado para la autoridad de certificación que produjo el siguiente. Es funcionalmente equivalente al siguiente certificado $X_1\langle\langle X_{n+1}\rangle\rangle$. Por ejemplo la posesión de A<<B>>B<<C>> confiere la misma capacidad que A<<C>>, a saber, la aptitud para hallar Cp cuando se da Ap |
| $X_{1p} \cdot X_1\langle\langle X_2\rangle\rangle$ | la operación de desenvolver un certificado (o cadena de certificados) para extraer una clave pública. Es un operador infijo, cuyo operando izquierdo es la clave pública de una autoridad de certificación, y cuyo operando derecho es un certificado emitido por esa autoridad de certificación. El resultado es la clave pública del usuario cuyo certificado es el operando derecho. Por ejemplo:<br>$A_p$ .A<<B>>B<<C>><br>denota la operación de usar la clave pública de A para obtener la clave pública de B, $B_p$, de su certificado, seguido por el uso de $B_p$ para desenvolver el certificado de C. El resultado de la operación es la clave pública de C, $C_p$ |
| A→B | un trayecto de certificación de A a B, formado por una cadena de certificados, que comienza por AC(A)<<$AC^2$(A)>> y termina por AC(B)<<B>>. |

## 5 Procedimiento de autenticación simple

5.1 La autenticación simple tiene por objeto proporcionar una autorización local basada en un nombre distinguido de usuario, una contraseña (opcional) convenida bilateralmente y un entendimiento mutuo sobre los medios para utilizar y tratar esta contraseña dentro de un solo dominio. La utilización de la autenticación simple tiene como finalidad inicial el uso local solamente, es decir, a la autenticación de entidades pares entre un AUG y un ASG, o entre un ASG y otro ASG. La autenticación simple puede efectuarse de varios modos:

a) la transferencia del nombre distinguido del usuario y la contraseña (opcional) en lenguaje ordinario (no protegido) al destinatario, para su evaluación;

b) la transferencia del nombre distinguido del usuario, la contraseña, y un número aleatorio y/o una indicación de tiempo, todo lo cual se protege mediante la aplicación de una función unidireccional;

c) la transferencia de la información protegida descrita en b) junto con un número aleatorio y/o una indicación de tiempo, todo lo cual se protege por la aplicación de una función unidireccional.

*Nota 1* - No se exige que las funciones unidireccionales aplicadas sean diferentes.

*Nota 2* - Los procedimientos de señalización para proteger las contraseñas pueden ser una cuestión de interés para la ampliación de la Recomendación.

5.2 Cuando las contraseñas no están protegidas, se proporciona un mínimo grado de seguridad para impedir un acceso no autorizado. Esto no debe considerarse una base para servicios seguros. La protección del nombre distinguido y de la contraseña del usuario da un mayor grado de seguridad. Los algoritmos para uso en el mecanismo de protección son, típicamente, funciones unidireccionales no cifrantes, que son muy fáciles de implementar.

5.3 El procedimiento general para la obtención de una autenticación simple se muestra en la figura 1/X.509.

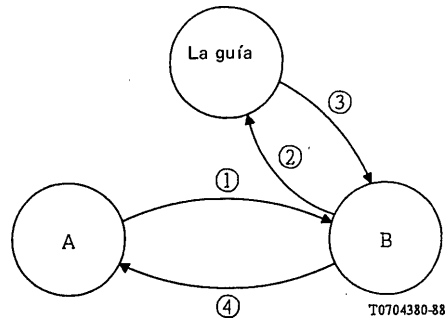



FIGURA 1/X.509

**Procedimiento de autenticación simple no protegida**

5.3.1 Comprende los siguientes pasos:

1) un usuario originador A envía su nombre distinguido y contraseña a un usuario receptor (o destinatario) B;

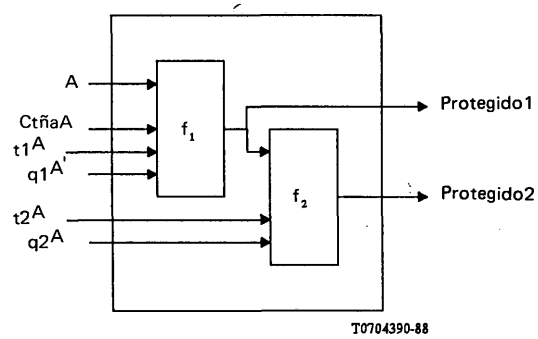
2) B envía el nombre distinguido contemplado y la contraseña de A a la guía, donde la contraseña se comprueba contra la mantenida como el atributo de **Contraseña de Usuario** dentro del asiento de la guía para A (usando la operación comparar de la guía);

3) la guía confirma (o rechaza) a B que las credenciales son válidas;

4) el éxito (o fracaso) de la autenticación puede comunicarse a A.

5.3.2 La forma básica de la autenticación simple comprende solamente el paso 1) y, después de que B ha verificado el nombre distinguido y la contraseña, puede incluir el paso 4).

5.4 La figura 2/X.509 muestra dos métodos que pueden emplearse para generar información de identificación protegida. $f_1$ y $f_2$ son funciones unidireccionales (que pueden ser idénticas o diferentes) y las indicaciones de tiempo y los números aleatorios son opcionales y están sujetos a acuerdo bilateral.




A = Nombre distinguido de usuario
$t^A$ = Indicaciones de tiempo
CtñaA = Contraseña de A
$q^A$ = Números aleatorios, y opcionalmente con un contador incluido

FIGURA 2/X.509

**Autenticación simple protegida**

5.4.1 La figura 3/X.509 ilustra el procedimiento de autenticación simple protegida.

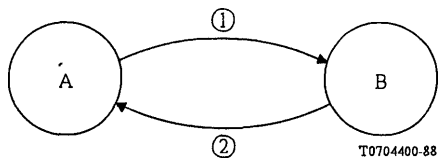



FIGURA 3/X.509

**El procedimiento de autenticación simple protegida**

Comprende los siguientes pasos (inicialmente sólo se utiliza $f_1$):

1) el usuario de origen, usuario A, envía su información de identificación protegida (Autenticador1) al usuario B. La protección se consigue aplicando la función unidireccional ($f_1$ de la figura 2/X.509, donde la indicación de tiempo y/o el número aleatorio (si se utiliza) tienen por finalidad minimizar la reproducción y ocultar la contraseña.

   La protección de la contraseña de A se realiza de la siguiente forma:

   Protegido1 = $f_1$ ($t1^A$; $q1^A$; Ctña A).

   La información transportada a B tiene la forma siguiente:

   Autenticador1 = $t1^A$; $q1^A$; A; Protegido1.

B verifica la información de identificación protegida ofrecida por A (utilizando para ello la indicación de tiempo, el nombre distinguido y, opcionalmente, la indicación de tiempo adicional y/o el número aleatorio proporcionado por A, junto con una copia local de la contraseña de A) y genera una copia protegida local de la contraseña de A (de la forma Protegido1). B compara (según el criterio de igualdad) la información de identificación contemplada (Protegido1) con el valor generado localmente.

2) B confirma (o rechaza) a A la verificación de la información de identificación protegida.

5.4.2 El procedimiento descrito en el § 5.4.1 puede modificarse para dar una mayor protección (mediante el empleo de $f_1$ y $f_2$). Las diferencias principales son las siguientes:

1) A envía su información de identificación protegida (adicionalmente) (Autenticación2) a B. Una protección adicional se obtiene aplicando una segunda función unidireccional, $f_2$, como se ilustra en la figura 2/X.509. Esta mayor protección adopta la forma siguiente:

Protegido2 = $f_2$ ($t2^A$, $q2^A$, Protegido1).

La información transportada a B tiene la forma:

Autenticador2 = $t1^A$, $t2^A$, $q1^A$, $q2^A$, A, Protegido2.

Para la comparación, B genera un valor local de la contraseña adicionalmente protegida de A y lo compara (según el criterio de igualdad) con el de Protegido2 (esto es similar, en principio al paso 1 del § 5.4.1);

2) B confirma (o rechaza) a A la verificación de la información de identificación protegida.

*Nota* - Los procedimientos definidos en esta cláusula se especifican sobre la base de A y B. Atendiendo a la aplicación a la guía (especificada en las Recomendaciones X.511 y X.518), A podría ser un AUG vinculado a un ASG, B; alternativamente A, podría ser un ASG vinculado a otro ASG, B.

5.5 Un tipo de atributo contraseña de usuario contiene la contraseña de un objeto. Un valor de atributo para la contraseña de usuario es una cadena especificada por el objeto.

```
UserPassword ::=  ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX
                    OCTET STRING (SIZE (0..ub-user-password))
                    MATCHES FOR EQUALITY
```

5.6 La siguiente macro NSA.1 puede utilizarse para definir el tipo datos que se obtiene al aplicar una función unidireccional a otro tipo dado de datos.

```
PROTECTEDMACRO ::= SIGNATURE
```

SECCION 2 - *Autenticación fuerte*

## 6 Bases de autenticación fuerte

6.1 El enfoque a la autenticación fuerte adoptado en esta Recomendación utiliza las propiedades de una familia de sistemas criptográficos, conocidos como criptosistemas de claves públicas (CSCP). Estos criptosistemas, también descritos como asimétricos implican un par de claves, una secreta y una pública, y no una sola clave, como los sistemas criptográficos convencionales. El anexo B da una breve introducción a estos criptosistemas y sus propiedades útiles para la autenticación. Para que un CSCP sea utilizable en este marco de autenticación, actualmente, debe tener la propiedad de que ambas claves del par de claves puedan ser usadas para el cifrado, empleándose la clave secreta para descifrar si se usó la clave pública, y empleándose la clave pública para descifrar si se usó la clave secreta. Dicho sea en otras palabras, $X_p \cdot X_s = X_s \cdot X_p$ siendo $X_p/X_s$ funciones de cifrado/descifrado que utilizan las claves pública/secreta de X.

*Nota* - En una futura y posible ampliación podrán especificarse otros tipos de CSCP, es decir, tipos que no requieran la propiedad de permutabilidad y que puedan ser soportados sin grandes modificaciones de esta Recomendación.

6.2 Este marco de autenticación no obliga a usar un criptosistema en particular. Se pretende que el marco sea aplicable a cualquier criptosistema de clave pública adecuado, y soportará por consiguiente cambios en los métodos usados como un resultado de futuros avances en criptografía, técnicas matemáticas o capacidades de computación. Sin embargo, dos usuarios que desean autenticar tienen que soportar el mismo algoritmo criptográfico para que la autenticación se realice correctamente. Así, dentro del contexto de un conjunto de aplicaciones relacionadas, la elección de un solo algoritmo servirá para maximizar la comunidad de usuarios capaces de autenticar y comunicar con seguridad. Un algoritmo criptográfico, que probablemente sea ampliamente usado, se especifica en el anexo C.

6.3 La autenticación se basa en que cada usuario posea un nombre distinguido único. La atribución de nombres distinguidos es responsabilidad de las autoridades de denominación. Cada usuario tiene por consiguiente que confiar en que las autoridades de denominación no expidan nombres distinguidos duplicados.

6.4 Cada usuario se identifica por su posesión de la clave secreta. Un segundo usuario puede determinar si su copartícipe en la comunicación está en posesión de la clave secreta, y puede usar esto para corroborar que su copartícipe en la comunicación es en realidad el usuario. La validez de esta corroboración depende de que la clave secreta permanezca confidencial para el usuario.

6.5 Para que un usuario determine que su copartícipe en la comunicación está en posesión de la clave secreta de otro usuario, deberá, él mismo, estar en posesión de la clave pública de ese usuario. Si bien la obtención del valor de esta clave pública a partir del asiento del usuario en la guía es inmediata, la verificación de su corrección plantea ciertos problemas. Puede haber varias formas posibles de realizar esto: el § 7 describe un proceso por el cual una clave pública de usuario puede ser verificada por referencia a la guía. Este proceso sólo puede operar si hay una cadena ininterrumpida de puntos de confianza, en la guía, entre los usuarios que solicitan autenticación. Esta cadena puede construirse identificando un punto común de confianza. Este punto común de confianza deberá estar enlazado con cada usuario por una cadena ininterrumpida de puntos de confianza.

## 7 Obtención de una clave pública de usuario

7.1 Para que un usuario confíe el procedimiento de autenticación, tiene que obtener la clave pública del otro usuario desde una fuente en la cual confía. Tal fuente, llamada autoridad de certificación (AC), usa el algoritmo de clave pública para certificar la clave pública, produciendo un *certificado.* El certificado, cuya forma se especifica en el § 7.2, tiene las siguientes propiedades:

- cualquier usuario con acceso a la clave pública de la autoridad de certificación puede extraer la clave pública que fue certificada;
- ninguna parte que no sea la autoridad de certificación puede modificar el certificado sin que esto sea detectado (los certificados son infalsificables).

Como los certificados son infalsificables, pueden publicarse insertándolos en la guía, sin que ésta tenga que tomar disposiciones especiales para protegerlos.

*Nota* - Aunque las AC están definidas inequívocamente por un nombre distinguido en el AIG, esto no implica que exista una relación entre la organización de las AC y el AIG.

7.2 Una autoridad de certificación produce el certificado de un usuario firmando (véase el § 8) una colección de informaciones, incluidos el nombre distinguido y la clave pública del usuario. Específicamente, el certificado de un usuario con el nombre distinguido A, producido por la autoridad de certificación AC, tiene la forma siguiente:

$$AC\langle\langle A\rangle\rangle = AC\ \{NS, IA, AC, A, Ap, T^A\}$$

donde NS es el número de serie de certificado, IA es el identificador del algoritmo utilizado para firmar el certificado, y $T^A$ indica el periodo de validez del certificado, y consiste en dos fechas, la primera y la última en las que el certificado es válido. Dado que se supone que $T^A$ se cambie en

periodos de no menos de 24 horas, se espera que los sistemas puedan usar el tiempo universal coordinado como una base de tiempo de referencia. La firma puede ser comprobada en cuanto a su validez por cualquier usuario que conozca ACP. El siguiente tipo de datos NSA.1 puede usarse para representar certificados:

```
Certificate     ::=  SIGNED SEQUENCE{
    version                 [0]Version DEFAULT 1988
    serialNumber            SerialNumber,
    signature               Algorithmidentifier
    issuer                  Name
    validity                Validity,
    subject                 Name,
    subjectPublicKeyInfo    SubjectPublicKeyInfo}

Version         ::=  INTEGER { 1988(0)}
SerialNumber    ::=  INTEGER

Validity        ::=
    SEQUENCE{
        notBefore  UTCTime,
        notAfter   UTCTime}

SubjectPublicKeyInfo ::=
    SEQUENCE{
        algorithm      AlgorithmIdentifier
        subjectKey     BIT STRING}

AlgorithmIdentifier  ::=
    SEQUENCE{
        algorithm      OBJECT IDENTIFIER
        parameters     ANY DEFINED BY algorithm
                           OPTIONAL}
```

7.3 El asiento de guía de cada usuario, A, que está participando en una autenticación fuerte, contiene el certificado (o los certificados) de A. Tal certificado es generado por una autoridad de certificación de A, que es una entidad en el AIG. Una autoridad de certificación de A, que puede no ser única, se denota por AC(A), o simplemente AC, si se sobreentiende A. La clave pública de A puede ser así descubierta por cualquier usuario que conoce la clave pública de AC. El descubrimiento de claves públicas es por tanto recursivo.

7.4 Si el usuario A, que trata de obtener la clave pública del usuario B, ya ha obtenido la clave pública de AC(B) el proceso habrá terminado. A fin de permitir que A obtenga la clave pública de AC(B), el asiento de guía de cada autoridad de certificación, X, contiene un número de certificados. Estos certificados son de dos tipos. En primer lugar, hay certificados de X en sentido de ida, denominado "directos" (forward), generados por otras autoridades de certificación. En segundo lugar, hay certificados de X en sentido de retorno, denominados "inversos" (reverse), generados por la propia X, los cuales son claves públicas certificadas de otras autoridades de certificación. La existencia de estos certificados permite a los usuarios construir trayectos de certificación de un punto a otro.

7.5 La lista de los certificados que se necesitan para permitir a un determinado usuario descubrir la clave pública de otro se conoce como el *trayecto de certificación*. Cada ítem en la lista es un certificado de la autoridad de certificación para la siguiente. Un trayecto de certificación de A a B (designado por A → B):

- comienza por el certificado inverso producido por AC(A), a saber $AC(A)\langle\langle X^1\rangle\rangle$ para alguna entidad $X^1$;
- continúa con ulteriores certificados $X^i\langle\langle X^{i+1}\rangle\rangle$;
- finaliza con el certificado de B.

Un trayecto de certificación forma lógicamente una cadena ininterrumpida de puntos de confianza en el árbol de información de la guía, entre dos usuarios que desean autenticar. El método preciso empleado por los usuarios A y B para obtener trayectos de certificación A → B y B → A puede variar. Una manera de facilitar esto consiste en organizar una jerarquía de ACs, que puede o no coincidir con la totalidad o una parte de la jerarquía del AIG. La ventaja de esto es que los usuarios que tienen ACs en la jerarquía pueden establecer entre sí un trayecto de certificación utilizando la guía sin ninguna información previa; para que esto sea posible, cada AC puede almacenar un certificado (directo) y un certificado inverso designado como correspondiente a su AC superior.

7.6 Los certificados están contenidos en asientos de la guía como atributos de tipo **certificado usuario, certificado AC** y **par de certificados cruzados.** Estos tipos de atributos son conocidos por la guía. Se puede actuar sobre estos atributos utilizando las mismas operaciones de protocolo empleadas para atributos. La definición de estos tipos puede encontrarse en el § 3.3 de esta Recomendación; la especificación de estos tipos de atributo es la siguiente:

```
UserCertificate          ::=    ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Certificate

CACertificate            ::=    ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Certificate

CrossCertificatePair     ::=    ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX CertificatePair

CertificatePair          ::=
    SEQUENCE{
        forward [0] Certificate OPTIONAL
        reverse [1] Certificate OPTIONAL
        -- por lo menos uno debe estar presente --}
```

Un usuario puede obtener uno o más certificados de una o más autoridades de certificación. Cada certificado comporta el nombre de la autoridad de certificación que lo expidió.

7.7 En el caso general, antes de que los usuarios puedan autenticar mutuamente, la guía tiene que suministrar los trayectos completos de certificación, y de certificación de retorno. Sin embargo, en la práctica, la cantidad de información que hay que obtener de la guía para una instancia particular de autenticación se puede reducir por los medios siguientes:

a) si los usuarios que quieren autenticar son servidos por la misma autoridad de certificación, el trayecto de certificación resulta trivial y los usuarios desenvuelven directamente los certificados de cada uno de los otros;

b) un usuario pudiera almacenar claves públicas, certificados y certificados inversos de todas las autoridades de certificación entre el usuario y la raíz del AIG. Típicamente, esto entrañaría que el usuario conociera las claves públicas y los certificados de solamente tres o cuatro autoridades de certificación. El usuario sólo necesitaría entonces obtener los trayectos de certificación desde el punto común de confianza;

c) si un usuario se comunica frecuentemente con usuarios certificados por otra AC en particular, este usuario pudiera aprender el trayecto de certificación a ese AC y el trayecto de certificación de retorno desde ese AC, con lo que sólo sería necesario obtener el certificado del otro usuario, desde la guía;

d) las autoridades de certificación pueden certificarse mutuamente unas a otras, por acuerdos bilaterales. Como resultado de esto se acorta el trayecto de certificación.

e) si dos usuarios han comunicado antes y cada uno ha aprendido el certificado del otro, podrán autenticar sin recurrir a la guía.

De todas formas, los usuarios, después de haber conocido los certificados de cada uno de los demás en base al trayecto de certificación, deberán verificar la validez de los certificados recibidos.

7.8 (Ejemplo). La figura 4/X.509 ilustra un ejemplo hipotético de un fragmento del AIG, en el cual las AC forman una jerarquía. Además de la información indicada en las AC, se supone que cada usuario conoce la clave pública de su autoridad de certificación, y sus propias claves pública y secreta.

7.8.1 Si las AC de los usuarios forman una jerarquía, A puede obtener los siguientes certificados de la guía para establecer un trayecto de certificación a B:

X<<W>>, W<<V>>, V<<Y>>, Y<<Z>>, Z<<B>>.

Una vez que A ha obtenido estos certificados, puede desenvolver secuencialmente el trayecto de certificación para obtener el contenido del certificado de B, incluido Bp:

Bp = Xp · X<<W>> W<<V>> V<<Y>> Y<<Z>> Z<<B>>.

En general A tiene también que adquirir de la guía los siguientes certificados para establecer el trayecto de certificación de retorno de B a A:

Z<<Y>>, Y<<V>>, V<<W>>, W<<X>>, X<<A>>.

Cuando B recibe estos certificados desde A, puede desenvolver secuencialmente el trayecto de certificación de retorno para obtener el contenido del certificado de A, incluido Ap:

Ap = Zp · Z<<Y>> Y<<V>> V<<W>> W<<X>> X<<A>>.

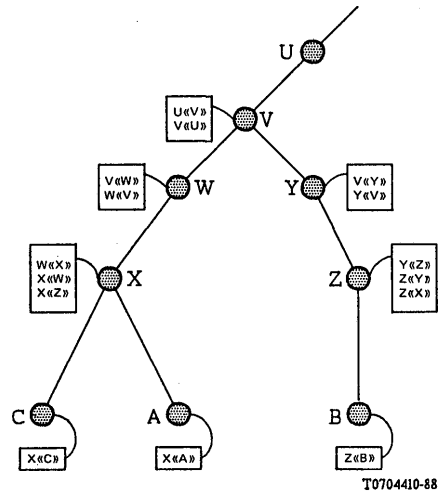



FIGURA 4/X.509

**Jerarquía de AC - Un ejemplo hipotético**

7.8.2 Aplicando las optimizaciones del § 7.7:

a) tomando A y C, por ejemplo: ambos conocen Xp, de modo que, sencillamente, A tiene que adquirir directamente el certificado de C. El desenvolvimiento del trayecto de certificación se reduce a:

Cp = Xp · X<<C>>

y el desenvolvimiento del trayecto de certificación de retorno se reduce a:

Ap = Xp · X<<A>>.

b) suponiendo que A conociera así W<<X>>, Wp, V<<W>>, Vp, U<<V>>, hacia arriba, etc., la información que A tiene que obtener de la guía para formar el trayecto de autenticación se reduce a:

V<<Y>>, Y<<Z>>, Z<<B>>

y la información que A tiene que obtener de la guía para formar el trayecto de certificación de retorno se reduce a:

Z<<Y>>, Y<<V>>.

c) suponiendo que A comunica frecuentemente con usuarios certificados por Z, él puede aprender (además de las claves públicas aprendidas en b)) V<<Y>>, Y<<V>>, Y<<Z>>, y Z<<Y>>. Para comunicar con B, sólo necesita por consiguiente obtener Z<<B>> de la guía;

d) suponiendo que los usuarios certificados por X y Z comunican frecuentemente, entonces X<<Z>> estaría contenido en el asiento de la guía para X, y viceversa (esto se muestra en la figura 4/X.509). Si A quiere autenticar hacia B, sólo necesita obtener:

X<<Z>>, Z<<B>>

para formar el trayecto de certificación, y:

Z<<X>>

para formar el trayecto de certificación de retorno;

e) suponiendo que los usuarios A y C han comunicado antes y han aprendido sus certificados respectivos, cada uno puede usar directamente la clave del otro, por ejemplo:

Cp = Xp · X<<C>>

y

Ap = Xp · X<<A>>.

7.8.3 En el caso más general, las autoridades de certificación no guardan una relación jerárquica. En el ejemplo hipotético de la figura 5/X.509, supóngase que un usuario D, certificado por U, desea autenticar al usuario E, certificado por W. El asiento de guía del usuario D contendrá el certificado U<<D>> y el asiento del usuario E contendrá el certificado W<<E>>.

Sea V una AC con la cual las AC, U y W han efectuado anteriormente cierto intercambio de redes públicas en una situación de confianza. Como resultado de esto se han generado y almacenado en la guía, certificados U<<V>>, V<<U>>, W<<V>> y V<<W>>. Supóngase que U<<V>> y W<<V>> están almacenados en el asiento de V, V<<U>> está almacenado en el asiento de U, y V<<W>> está almacenado en el asiento de W.

El usuario D debe encontrar un trayecto de certificación E. Este usuario podría utilizar diversos métodos. Uno de ellos consistiría en considerar los usuarios y ACs como nodos, y los certificados como arcos en un gráfico dirigido. En estos términos, D debe efectuar una búsqueda en el gráfico para encontrar un trayecto de U a E, siendo uno de ellos U<<V>>, V<<W>>, W<<E>>. Una vez descubierto este trayecto, se puede construir también el trayecto inverso W<<V>>, V<<U>>, U<<D>>.

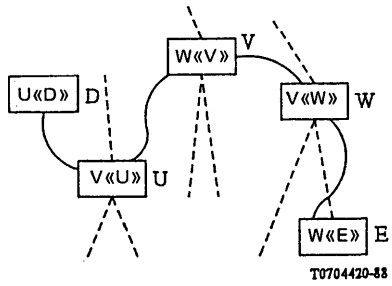



FIGURA 5/X.509

**Ejemplo de trayecto de certificación no jerárquico**

## 8 Firmas digitales

En esta sección no se pretende especificar una norma para firmas digitales en general, sino especificar los medios para firmar los testigos en la guía.

8.1 La información (info) se firma añadiéndole un sumario cifrado de la información. El sumario se produce por medio de una función hash unidireccional, mientras que el cifrado se lleva a cabo usando la clave secreta del firmante (véase la figura 6/X.509). Así

X{Info} = Info,Xs[h (Info)]

*Nota* - El cifrado mediante la clave secreta asegura que la firma no puede ser falsificada. La naturaleza unidireccional de la función hash asegura que la información falsa, generada como para tener el mismo resultado hash (y por consiguiente la firma), no puede ser introducida en sustitución.

8.2 El receptor de información firmada verifica la firma:

- aplicando la función hash unidireccional a la información;
- comparando el resultado con el obtenido descifrando la firma mediante la clave pública del firmante.

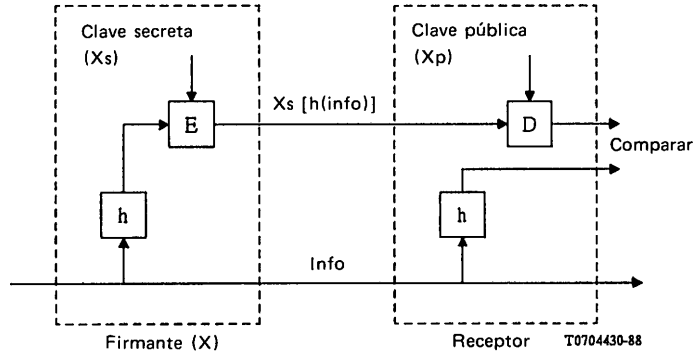



FIGURA 6/X.509

**Firmas digitales**

8.3 Este marco de autenticación no impone una sola función hash unidireccional para uso en firmado. Se pretende que el marco sea aplicable a cualquier función hash adecuada, y que por consiguiente admita cambios de los métodos usados, como un resultado de futuros avances en criptografía, técnicas matemáticas o capacidades de computación. Sin embargo, dos usuarios que quieran autenticar tienen que soportar la misma función hash para que la autenticación se realice correctamente. Por consiguiente, dentro del contexto de un conjunto de aplicaciones relacionadas, la elección de una sola función servirá para maximizar la comunidad de usuarios capaces de autenticar y comunicar con seguridad. Una función hash que tiene probabilidad de ser ampliamente usada se especifica en el anexo D.

La información firmada incluye indicadores que identifican el algoritmo de la función hash y el algoritmo de encripción utilizados para computar la firma digital.

8.4 El cifrado de algún ítem de datos puede describirse utilizando la siguiente macro NSA.1:

```
ENCRYPTED MACRO    ::=
BEGIN

TYPE NOTATION      ::= type (ToBeEnciphered)
VALUE NOTATION     ::= value (VALUE BIT STRING)
END
```

El valor de la cadena de bits se genera tomando los octetos que forman la codificación completa (utilizando las reglas de codificación básica NSA.1) del valor del tipo A-Cifrar (**ToBeEnciphered**) y aplicando un procedimiento de cifrado a esos octetos.

*Nota 1* - El procedimiento de encripción requiere un acuerdo sobre el algoritmo a aplicar, incluyendo los eventuales parámetros de algoritmo, así como toda clave, valor de inicialización e instrucción de relleno que pueda necesitarse. Es en los procedimientos de encripción donde se especificarán los medios para obtener la sintonización de los datos de emisor y del receptor, lo que puede incluir información en los bits que deban transmitirse.

*Nota 2* - El procedimiento de encripción deberá admitir como entrada una cadena de octetos y generar una cadena única de bits, como resultado.

*Nota 3* - El mecanismo para el acuerdo de seguridad sobre el algoritmo de encripción y sus parámetros, el emisor y el receptor de los datos, están fuera del ámbito de esta Recomendación.

8.5 Cuando deba asociarse una firma a un tipo de datos, puede utilizarse la siguiente macro NSA.1 para definir el tipo de datos resultantes de la aplicación de una firma a un determinado tipo de datos.

```
SIGNED MACRO      ::=
BEGIN

TYPE NOTATION     ::= type (ToBeSigned)

VALUE NOTATION    ::= value (VALUE

    SEQUENCE{
        ToBeSigned,
        AlgorithmIdentifier,
        -- del algoritmo utilizado
        -- para computar la firma
        ENCRYPTED OCTET STRING
        -- donde la cadena de octetos
        -- es el resultado de aplicar
        -- la función hash del valor de
        -- 'ToBeSigned' --}

END -- of SIGNED       )
```

8.6 Cuando sólo se requiera la firma, puede utilizarse la siguiente macro NSA.1 para definir el tipo de datos resultante de la aplicación de una firma al tipo de datos dado.

```
SIGNATURE MACRO   ::=
BEGIN

TYPE NOTATION     ::= type (OfSignature)
VALUE NOTATION    ::= value (VALUE

    SEQUENCE{
        AlgorithmIdentifier,
        -- del algoritmo utilizado
        -- para computar la firma
        ENCRYPTED OCTET STRING
        -- donde la cadena de octetos es una función
        -- (por ejemplo, una versión comprimida o tratada
        -- por la función hash) del valor 'OfSignature',
        -- que puede incluir el identificador del algoritmo
        -- utilizado para computar la firma --}

END -- of SIGNATURE    )
```

8.7 A fin de permitir la validación de los tipos **SIGNED** y **SIGNATURE** en un entorno distribuido, se requiere codificación distinguida. Una codificación distinguida de un valor de datos **SIGNED** o **SIGNATURE** se obtendrá aplicando las Reglas de Codificación Básicas definidas en la Recomendación X.209 con las siguientes limitaciones:

a) se utilizará la forma definida de codificación de longitud, codificada en el mínimo número de octetos;

b) para los tipos cadena, no se utilizará la forma construida de codificación;

c) si el valor de un tipo es su valor por defecto, deberá estar ausente;

d) los componentes de un tipo Conjunto deberán codificarse en orden ascendente de su valor de rótulo;

e) los componentes de un tipo Conjunto-de se codificarán en orden ascendente de su valor de octeto;

f) si el valor de un tipo Booleano es verdadero, el octeto de contenido de la codificación deberá fijarse a $'FF'_{16}$ ;

g) todo bit no utilizado en el octeto final de la codificación de un valor Cadena de Bits, si existe, deberá fijarse a cero;

h) el tipo Real se codificará de una manera tal que no se utilicen las bases 8, 10 y 16, y el factor binario de afectación en escala será cero.

## 9 Procedimiento de autenticación fuerte

### 9.1 *Visión de conjunto*

9.1.1 El enfoque básico de la autenticación se ha resumido anteriormente, esto es: corroborar la identidad demostrando la posesión de una clave secreta. Sin embargo, son posibles muchos procedimientos de autenticación que emplean este enfoque. En general incumbe a una aplicación específica el determinar los procedimientos apropiados, de modo que se cumpla su política de seguridad. Esta cláusula describe tres procedimientos distintos de autenticación, que quizás resulten útiles en una gama de aplicaciones.

*Nota* - Esta Recomendación no especifica los procedimientos con el detalle requerido para la implementación. Sin embargo, pueden preverse normas adicionales que lo hicieran, sea de una manera específica a la aplicación o en un modo de propósito general.

9.1.2 Los tres procedimientos comprenden diferentes números de intercambios de información de autenticación, y en consecuencia, proporcionan diferentes tipos de seguridades a los participantes. Específicamente,

a) la autenticación unidireccional, descrita en el § 9.2 implica una transferencia simple de información desde un usuario (A) prevista para otro (B), y determina lo siguiente:
   - la identidad de A, y que el testigo de autenticación fue generado realmente por A;
   - la identidad de B, y que el testigo de autenticación se previó realmente enviarlo a B;
   - la integridad y 'originalidad' (la propiedad de no haber sido enviado dos o más veces) del testigo de autenticación que está siendo transferido.

   Las últimas propiedades pueden ser determinadas también para todo otro dato arbitrario adicional en la transferencia;

b) la autenticación bidireccional, descrita en el § 9.3, implica, además, una respuesta de B a A. Determina además lo siguiente:
   - que el testigo de autenticación generado en la respuesta fue generado realmente por B y estaba previsto para ser enviado a A;
   - la integridad y originalidad del testigo de autenticación enviado en la respuesta;
   - (opcionalmente), el secreto mutuo de una parte de los testigos;

c) la autenticación tridireccional, descrita en el § 9.4, implica, además, una transferencia ulterior de A a B. Determina las mismas propiedades que la autenticación bidireccional, pero lo hace sin necesidad de comprobación de la indicación de la hora de la asociación.

En cada caso donde va a tener lugar una autenticación fuerte, A tiene que obtener la clave pública de B y el trayecto de certificación de retorno de B a A, previamente a cualquier intercambio de información. Esto puede implicar acceso a la guía, como se describió en el § 7 anteriormente. Tal tipo de acceso no se vuelve a mencionar en la descripción de los procedimientos que siguen.

La comprobación de las indicaciones de hora mencionadas en las siguientes secciones solamente es aplicable cuando, o bien se usan relojes sincronizados en un entorno local, o cuando los relojes están sincronizados lógicamente por acuerdos bilaterales. En cualquier caso, se recomienda que se use el tiempo universal coordinado.

En cada uno de los procedimientos de autenticación descritos a continuación se supone que la parte A ha comprobado la validez de todos los certificados en el trayecto de certificación.

### 9.2 *Autenticación unidireccional*

Se siguen los siguientes pasos que muestra la figura 7/X.509:

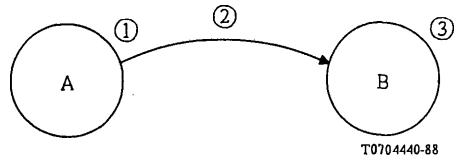



FIGURA 7/X.509

**Autenticación unidireccional**

1) A genera $r^A$, un número no repetitivo, que se usa para detectar ataques de reactuación y para prevenir la falsificación.

2) A envía el siguiente mensaje a B:

   $B \rightarrow A, A\{t^A, r^A, B\}$

   donde $t^A$ es una indicación de tiempo. $t^A$ consta de una o dos fechas: la hora de generación del testigo (que es facultativa) y la fecha de expiración. Como otra posibilidad, si se debe proporcionar autenticación del origen de datos de 'sgnData' por firma digital:

   $B \rightarrow A, A\{t^A, r^A, B, sgnData\}$

   En los casos en que haya que transportar información que vaya a utilizarse posteriormente como una clave secreta (a dicha información se le llama 'encData'):

   $B \rightarrow A, A\{t^A, r^A, B, sgnData, Bp[encData]\}$.

   La utilización de 'encData' como una clave secreta implica que deberá elegirse ésta con cuidado; por ejemplo, deberá procurarse que sea una clave fuerte para cualquier criptosistema utilizado, como se indica en el campo 'sgnData' del testigo.

3) B efectúa las acciones siguientes:

   a) obtiene Ap de $B \rightarrow A$, comprobando que el certificado de A no ha expirado;

   b) verifica la firma, y por consiguiente la integridad de la información firmada;

   c) comprueba que él mismo (B) es el receptor deseado;

   d) comprueba que la indicación de tiempo está actual;

   e) opcionalmente, comprueba que $r^A$ no ha sido maniobrada. Esto pudiera lograrse, por ejemplo, haciendo que $r^A$ incluya una parte secuencial que es comprobada por una implementación local para detectar que su valor es único.

$r^A$ es válido hasta la fecha de expiración indicada por $t^A$. $r^A$ va siempre acompañado por una parte secuencial, que indica que A no repetirá el testigo durante el intervalo de tiempo $t^A$, y por tanto que no es necesaria la verificación del valor de $r^A$ propiamente dicho.

En todo caso, es razonable para B almacenar la parte secuencial junto con la indicación de hora $t^A$ en lenguaje ordinario junto con la parte a que se aplicó la función hash del testigo durante el intervalo de tiempo $t^A$.

### 9.3 *Autenticación bidireccional*

Se siguen los pasos indicados en la figura 8/X.509.

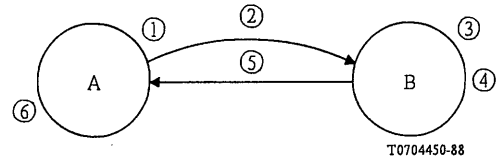



FIGURA 8/X.509

**Autenticación bidireccional**

1) Como en el § 9.2.
2) Como en el § 9.2.
3) Como en el § 9.2.
4) B genera $r^B$, un número no repetitivo, utilizado para fines similares a los de $r^A$.
5) B envía el siguiente testigo de autenticación a A:

   $B\{t^B, r^B, A, r^A\}$

   donde $t^B$ es una indicación de tiempo definida de la misma manera que $t^A$.

   Como otra posibilidad, si debe proporcionarse autenticación de origen de datos de 'sgnData' por firma digital:

   $B\ \{t^B, r^B, A, r^A,$ sgnData$\}$.

   En los casos en que haya que transportar información que vaya a utilizarse posteriormente como una clave secreta (a dicha información se le llama 'enData'):

   $B\ \{t^B, r^B, A, r^A,$ sgnData, Ap[encData]$\}$

   La utilización de 'encData' como clave secreta implica que deberá elegirse con cuidado; por ejemplo deberá ser una clave fuerte para cualquier criptosistema que se utilice en el campo 'sgnData' del testigo.

6) A ejecuta las siguientes acciones:
   a) verifica la firma, y por tanto la integridad de la información firmada;
   b) comprueba que A es el receptor deseado;
   c) comprueba que la indicación de hora $t^B$ es 'corriente';
   d) opcionalmente, comprueba que $r^B$ no ha sido maniobrado [véase el § 9.2 paso 3 e)].

9.4 *Autenticación tridireccional*

Se siguen los siguientes pasos indicados en la figura 9/X.509:

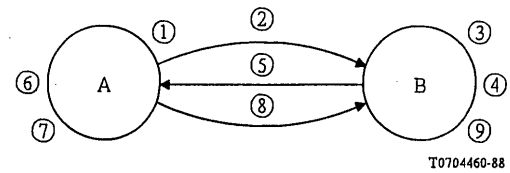



FIGURA 9/X.509

**Autenticación tridireccional**

1) Como en el § 9.3.

2) Como en el § 9.3. La indicación de tiempo $t^A$ puede ser cero.

3) Como en el § 9.3, excepto que la indicación de tiempo no necesita ser comprobada.

4) Como en el § 9.3.

5) Como en el § 9.3. La indicación de tiempo $t^B$ puede ser cero.

6) Como en el § 9.3, excepto que la indicación de tiempo no necesita ser comprobada.

7) A comprueba que el $r^A$ recibido es idéntico al $r^A$ que fue enviado.

8) A envía el siguiente testigo de autenticación a B:

   $A\{r^B\}$.

9) B efectúa las siguientes acciones:

   a) descifra el testigo de autenticación, después comprueba la firma y por consiguiente la integridad de la información firmada;

   b) comprueba que el $r^B$ recibido es idéntico al $r^B$ que fue enviado por B.

## 10 Gestión de claves y certificados

### 10.1 *Generación de pares de claves*

10.1.1 La política en general de gestión de seguridad de una implementación definirá el ciclo de vida de los pares de claves, y está por consiguiente fuera del alcance del marco de autenticación. Sin embargo, es vital a la seguridad general que todas las claves secretas permanezcan secretas, es decir, sólo conocidas por el usuario a que pertenecen.

Los datos de la clave no son fáciles de recordar por el usuario humano, por lo que hay que emplear un método apropiado para almacenarla en un modo transportable conveniente. Un mecanismo posible sería el uso de una "tarjeta inteligente" que podría contener las claves secreta y pública del usuario, el certificado del usuario, y una copia de la clave pública de la autoridad de certificación. El uso de esta tarjeta podría también asegurarse por medio de un NIP (número de identificación personal), que aumenta la seguridad del sistema al requerir del usuario la posesión de la tarjeta y que sepa cómo acceder al sistema. El método exacto escogido para almacenar tales datos, sin embargo, está fuera del ámbito de esta Recomendación.

10.1.2 Hay tres modos en los cuales un par de claves del usuario pueden ser producidos como se describe en el § 10.1.2.1 a 10.1.2.3.

10.1.2.1 El usuario genera su propio par de claves. Este método tiene la ventaja de que una clave secreta del usuario nunca es pasada a otra entidad, pero requiere un cierto nivel de competencia por el usuario, como se describe en el anexo C.

10.1.2.2 El par de claves es generado por una tercera entidad. La tercera entidad tiene que pasar la clave secreta al usuario de una manera físicamente segura, y entonces destruir activamente toda la información relacionada a la creación del par de claves más las propias claves. Hay que emplear medidas de seguridad adecuadas para garantizar que la tercera entidad y las operaciones de datos no son objeto de fraudes.

10.1.2.3 El par de claves se genera por la AC. Este es un caso especial del § 10.1.2.2 y las consideraciones hechas allí son aplicables.

*Nota* - La autoridad de certificación ya presenta funcionalidad fiduciaria con respecto al usuario, y estará sujeta a las medidas necesarias de seguridad física. Este método tiene la ventaja de no requerir una transferencia securizada de datos a la AC para la certificación.

10.1.2.4 El criptosistema en uso impone restricciones (técnicas) particulares a la generación de claves.

### 10.2 *Gestión de certificados*

10.2.1 Un certificado asocia la clave pública y el nombre distinguido único del usuario que el mismo describe. Por consiguiente:

a) una autoridad de certificación tiene que estar satisfecha de la identidad de un usuario antes de crear un certificado para el mismo;

b) una autoridad de certificación no expedirá certificados para dos usuarios con el mismo nombre.

10.2.2 La producción de un certificado ocurre fuera de línea y no deberá realizarse con un mecanismo automático de pregunta/respuesta. La ventaja de esta certificación es que debido a la clave secreta de la autoridad de certificación, la AC, nunca se conoce excepto en la AC aislada y físicamente segura, la AC secreta sólo puede ser entonces averiguada por un ataque a la propia AC, lo que hace poco probable un compromiso.

10.2.3 Es importante que la transferencia de información a la autoridad de certificación no sea comprometida, y hay que tomar medidas de seguridad física adecuadas. A este respecto:

a) se produciría una seria brecha en la seguridad si la AC expidiera un certificado para un usuario con una clave pública que haya sido objeto de un fraude;

b) si se emplea el medio de generación de los pares de claves del § 10.1.2.3, no se necesita una transferencia segura;

c) si se emplea el medio de generación de pares claves descrito en los § 10.1.2.1 ó 10.1.2.2, el usuario puede emplear diferentes métodos (en línea o fuera de línea) para comunicar su clave pública a la AC de una manera segura. Los métodos en línea pueden proporcionar una mayor flexibilidad para las operaciones a distancia efectuadas entre el usuario y la AC.

10.2.4 Un certificado es una información disponible públicamente, y no se necesita emplear medidas de seguridad específicas con respecto a su transporte a la guía. Como éste es producido por una autoridad de certificación "fuera de línea" a nombre de un usuario que recibirá una copia del mismo, el usuario necesita solamente almacenar esta información en su asiento de la guía en un acceso ulterior a la guía. Alternativamente la AC podría custodiar el certificado para el usuario, en cuyo caso a este agente tendrían que otorgársele derechos de acceso adecuados.

10.2.5 Los certificados tendrán asociada cierta duración, al final de la cual caducan (expiran). A fin de asegurar la continuidad del servicio, la AC garantizará el suministro oportuno de certificados de sustitución que reemplazan a los caducados o próximos a caducar. Los distintos aspectos de esta cuestión se describen en los § 10.2.5.1 y 10.2.5.2.

10.2.5.1 La validez de los certificados deberá organizarse de tal modo que la validez de uno entrañe la caducidad del precedente, o se puede permitir que sus periodos de validez se superpongan. Esto último evita que las AC tengan que instalar y distribuir un gran número de certificados que pudieran agotarse en la misma fecha de expiración.

10.2.5.2 Los certificados caducos normalmente serán sacados de la guía. Es cuestión de política de seguridad y de responsabilidad de la AC mantener los antiguos certificados durante cierto periodo de tiempo si no se presta el servicio de "incuestionabilidad de los datos" (denominado también "no-repudio de los datos").

10.2.6 Los certificados pueden ser revocados antes de su expiración, por ejemplo si se supone que la clave secreta del usuario puede ser objeto de maniobras irregulares, o si el usuario ya no deberá ser certificado por la AC, o si se supone que el certificado de la AC ha sido objeto de maniobras irregulares. Los distintos aspectos de esta cuestión se describen en los § 10.2.6.1 a 10.2.6.4.

10.2.6.1 La revocación de un certificado de usuario o de un certificado de AC debe ponerse en conocimiento de la AC, y deberá expedirse un nuevo certificado si fuese procedente. La AC podrá entonces informar al propietario del certificado, sobre su revocación, por un procedimiento fuera de línea.

10.2.6.2 La AC mantendrá:

a) una lista, con indicación de tiempo, de los certificados expedidos que han sido revocados;

b) una lista, con indicación de tiempo, de los certificados revocados de todas las AC, conocidos por la AC, certificados por la AC.

Ambas listas certificadas existirán, incluso si estuviesen vacías.

10.2.6.3 El mantenimiento de asientos de la guía afectados por las listas de revocaciones, por la AC, es responsabilidad de la guía y sus usuarios, quienes actúan de acuerdo con la política de seguridad. Por ejemplo, el usuario puede modificar su asiento de objeto reemplazando el antiguo certificado por uno nuevo. Este último se utilizará entonces para autenticar el usuario ante la guía.

10.2.6.4 Las listas de revocaciones ("listas negras") se mantienen dentro de asientos como atributos de tipos **"lista de revocaciones de certificados"** y **"lista de revocaciones de autoridad"**. Esos atributos pueden ser operados utilizando los mismos procedimientos empleados para otros atributos. Estos tipos de atributo se definen como sigue:

```
CertificateRevocationList    ::=    ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX CertificateList

AuthorityRevocationList      ::=    ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX CertificateList

CertificateList              ::=    SIGNED SEQUENCE{
    signature AlgorithmIdentifier,
    issuer Name,
    lastUpdate UTCTime,
    revokedCertificates
        SIGNED SEQUENCE OF SEQUENCE{
            signature AlgorithmIdentifier,
            issuer Name, CertificateSerialNumber subject,
            revocationDate UTCTime}
                OTPIONAL}
```

*Nota 1* - La verificación de la totalidad de los certificados es un asunto local.

*Nota 2* - Si un servicio de "incuestionabilidad de los datos" depende de claves proporcionadas por la AC, dicho servicio deberá asegurar que todas las claves pertinentes de la AC (revocadas o caducas) y las listas de revocaciones con indicación de tiempo son archivadas y certificadas por una autoridad "corriente".

## ANEXO A

(a la Recomendación X.509)

**Requisitos de seguridad**

Este anexo no forma parte integrante de esta Recomendación.

[Material adicional sobre este tema puede encontrarse en la Norma ISO 7498 - Information Proccessing Systems - OSI Reference Model - Part 2, Security Architecture.]

Muchas aplicaciones ISA, servicios definidos por el CCITT y servicios no definidos por el CCITT tendrán requisitos o exigencias de seguridad. Tales requisitos se derivan de la necesidad de proteger la transferencia de información contra una serie de peligros potenciales.

### A.1 *Peligros*

Algunos peligros comúnmente conocidos son:

a) *Intercepción de identidad:* la identidad de uno o más de los usuarios que participan en una comunicación se observa con el fin de detectar todo uso incorrecto.

b) *Impostura (o mascarada):* pretensión de un usuario de ser otro diferente para ganar acceso a información u obtener privilegios adicionales.

c) *Reactuación:* grabación y posterior reactuación de una comunicación en alguna fecha posterior.

d) *Intercepción de datos:* observación de datos de un usuario durante una comunicación, por un usuario no autorizado.

e) *Manipulación:* reemplazo, inserción, eliminación u ordenación incorrecta de datos de usuario durante una comunicación, por un usuario no autorizado.

f) *Repudio:* negación por un usuario de haber participado en una comunicación, o parte de ella.

g) *Denegación de servicio:* prevención o interrupción de una comunicación o la demora de operaciones críticas en cuanto al tiempo.

*Nota* - Este peligro para la seguridad es más general y depende de la aplicación individual o de la intención de la perturbación no autorizada y por consiguiente no está explícitamente dentro del ámbito del marco de autenticación.

h) *Encaminamiento incorrecto:* encaminamiento incorrecto de un trayecto de comunicación previsto de un usuario a otro.

*Nota* - El encaminamiento incorrecto ocurrirá naturalmente en las capas 1 a 3 de ISA, por lo que está fuera del ámbito del marco de autenticación. Sin embargo, puede ser posible evitar las consecuencias del encaminamiento incorrecto utilizando servicios apropiados de seguridad como los suministrados dentro del marco de autenticación.

i) *Análisis de tráfico:* observación de información sobre una comunicación entre usuarios (por ejemplo, ausencia/presencia, frecuencia, dirección, secuencia, tipo, cantidad, etc.).

*Nota* - Los peligros de análisis de tráfico no están naturalmente limitados a una capa ISA determinada. Por consiguiente el análisis de tráfico está generalmente fuera del ámbito del marco de autenticación. Sin embargo, el análisis de tráfico puede ser protegido parcialmente generando tráfico adicional ininteligible (tráfico de relleno), usando datos aleatorios o cifrados.

## A.2 *Servicios de seguridad*

Para la protección contra los peligros conocidos deben prestarse diversos servicios de seguridad. Los servicios de seguridad suministrados por el marco de autenticación se realizan por medio del mecanismo de seguridad descrito en el § A.3 de este anexo.

a) *Autenticación de entidad par:* este servicio proporciona una corroboración de que un usuario en una determinada instancia de comunicación es el que se anuncia como tal. Pueden solicitarse dos servicios diferentes de autenticación de identidad par:

- *autenticación de entidad simple* (ya sea autenticación de entidad de *origen de datos* o autenticación de entidad de *receptor de datos*);
- *autenticación mutua*, donde ambos usuarios comunicantes se autentican el uno al otro.

Cuando se solicita un servicio de autenticación de entidad par, los dos usuarios acuerdan si sus identidades serán protegidas o no.

El servicio de autenticación de entidad par es soportado por el marco de autenticación. Puede ser usado para proteger contra la impostura y la reactuación, concernientes a las identidades de los usuarios.

b) *Control de acceso:* este servicio puede usarse para proteger contra el uso no autorizado de recursos. El servicio de control de acceso es proporcionado por la guía u otra aplicación y no es por consiguiente un asunto del marco de autenticación.

c) *Confidencialidad de datos:* este servicio puede usarse para suministrar protección de los datos contra una revelación no autorizada. El servicio de confidencialidad de datos está soportado por el marco de autenticación. El mismo puede usarse para proteger contra intercepción de datos.

d) *Integridad de datos:* este servicio suministra prueba de la integridad de los datos en una comunicación. El servicio de integridad de datos está soportado por el marco de autenticación. Puede usarse para detectar y proteger contra la manipulación.

e) *No-repudio:* este servicio suministra la prueba de la integridad y del origen de los datos -ambos en una relación infalsificable - que pueden ser verificados por cualquier tercero en cualquier momento.

A.3 *Mecanismos de seguridad*

Los mecanismos de seguridad que se describen aquí realizan los servicios de seguridad descritos en el § A.2.

a) *Intercambio de autenticación:* hay dos grados de mecanismos de autenticación suministrados por el marco de autenticación:

- *autenticación simple:* se basa en que el originador suministre su nombre y contraseña, los cuales son comprobados por el receptor;
- *autenticación fuerte:* se basa en el uso de técnicas criptográficas para proteger el intercambio de información de validación. En el marco de autenticación, la autenticación fuerte se basa en un esquema asimétrico.

El mecanismo de intercambio de autenticación se usa para soportar el servicio de autenticación de entidad par.

b) *Cifrado:* el marco de autenticación contempla el cifrado de datos durante la transferencia. Pueden usarse esquemas simétricos o asimétricos. El intercambio necesario de claves se realiza o bien dentro de un intercambio de autenticación precedente o "fuera de línea" en cualquier momento antes de la comunicación que se va a hacer. Este último caso está fuera del ámbito del marco de autenticación. El mecanismo de cifrado soporta el servicio de confidencialidad de datos.

c) *Integridad de los datos:* este mecanismo implica el cifrado de una cadena comprimida de los datos pertinentes a transmitir. Junto con los datos ordinarios, este mensaje se le envía al receptor. El receptor repite la compresión y el cifrado ulterior de los datos ordinarios y compara el resultado con el creado por el originador para probar la integridad.

El mecanismo de integridad de datos puede ser suministrado por el cifrado de los datos ordinarios comprimidos ya sea por un esquema asimétrico o por un esquema simétrico. (Con el esquema simétrico, la compresión y el cifrado de los datos pudieran ser procesados simultáneamente.) El mecanismo no es suministrado explícitamente por el marco de autenticación. Sin embargo, se suministra totalmente como una parte del mecanismo de firma digital (véase más adelante) usando un esquema asimétrico.

El mecanismo de integridad de datos soporta el servicio de integridad de datos. También soporta parcialmente el servicio de no-repudio (ese servicio también necesita el mecanismo de firma digital para que sus requisitos se cumplan plenamente).

d) *Firma digital:* este mecanismo implica el cifrado, por medio de la clave secreta del originador, de una cadena comprimida de los datos pertinentes que se van a transferir. La firma digital, junto con los datos ordinarios se envía al receptor. Similarmente al caso del mecanismo de integridad de datos, este mensaje se procesa por el receptor para probar la integridad. El mecanismo de firma digital también prueba la autenticidad del originador y la relación inequívoca entre el originador y los datos que se transfirieron.

El marco de autenticación soporta el mecanismo de firma digital usando un esquema asimétrico.

El mecanismo de signatura digital soporta el servicio de integridad de datos y también el servicio de no-repudio.

A.4 *Peligros contra los que protegen los servicios de seguridad*

La tabla al final de este anexo indica los peligros de seguridad contra los que cada servicio de seguridad puede proteger. La presencia de un asterisco (*) indica que un cierto servicio de seguridad ofrece protección contra cierto peligro.

A.5 *Negociación de servicios y mecanismos de seguridad*

La provisión de características de seguridad durante una instancia de comunicación requiere la negociación del contexto en el cual se requieren los servicios de seguridad. Esto implica el acuerdo en el tipo de mecanismos de seguridad y de parámetros que son necesarios para suministrar tales servicios de seguridad. Los procedimientos que se requieren para negociar los mecanismos y parámetros pueden o bien ser llevados a cabo como una parte integrante del procedimiento normal de establecimiento de conexión, o como un proceso separado. Los detalles precisos de estos procedimientos para la negociación no se especifican en este anexo.

| | SERVICIOS | | | |
|---|---|---|---|---|
| PELIGROS | Autenticación de entidad | Confidencialidad de datos | Integridad de datos | No-Repudio |
| Intercepción de Identidad | * (si se requiere) | | | |
| Intercepción de Datos | | * | | |
| Impostura | * | | | |
| Re-actuación | * (identidad) | | * (datos) | * |
| Manipulación | | | * | * |
| Repudio | | | | * |

## ANEXO B

(a la Recomendación X.509)

### Una introducción a la criptografía de claves públicas

Este anexo no forma parte integrante de esta Recomendación.

En los sistemas criptográficos convencionales, la clave usada para cifrar la información por el originador de un mensaje secreto es la misma usada por el receptor legítimo para descifrar el mensaje.

En los criptosistemas de claves públicas (CSCP), sin embargo, las claves vienen en pares; una de las cuales se usa para el cifrado y la otra para el descifrado. Cada par de claves se asocia con un usuario particular X. Una de las claves, conocida como la clave pública (Xp) se conoce públicamente, y puede ser usada por cualquier usuario para cifrar datos. Solamente X, quien posee la clave secreta complementaria (Xs), puede descifrar los datos. (Esto se representa por la notación D = Xs[Xp[D]]). Es computacionalmente irrealizable derivar la clave secreta a partir del conocimiento de la clave pública. Cualquier usuario puede entonces comunicar una información la cual solamente X puede hallar, cifrándola bajo Xp. Por extensión, dos usuarios pueden comunicar en secreto, usando cada uno la clave pública del otro para cifrar los datos, como se muestra en la figura B-1/X.509.

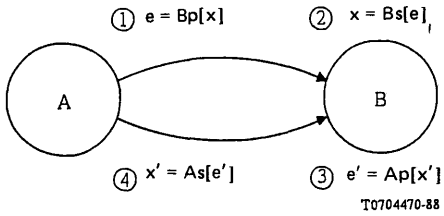



FIGURA B-1/X.509

**Uso de un CSCP para intercambiar información secreta**

El usuario A tiene la clave pública Ap y la clave secreta As, y el usuario B tiene otro conjunto de claves, Bp y Bs. A y B conocen cada uno la clave pública, pero no la clave secreta del otro. A y B pueden por consiguiente intercambiar información secreta entre ellos siguiendo los pasos siguientes (ilustrados en la figura B-1/X.509).

1) A desea enviar alguna información secreta x a B. A por consiguiente cifra x bajo la clave de cifrado de B y envía la información cifrada e a B. Esto se representa por:

   e = Bp[x]

2) B puede ahora descifrar este cifrado e para obtener la información x usando la clave secreta de descifrado Bs. Obsérvese que B es el único posesor de Bs, y debido a que esta clave puede que nunca sea revelada o enviada, es imposible para cualquier otra parte obtener la información x. La posesión de Bs determina la identidad de B. La operación de descifrado se representa por:

   x = Bs[e], o x = Bs[Bp[x]]

3) B puede ahora, análogamente, enviar alguna información secreta, x', a A, bajo la clave de cifrado de A, Ap:

   e' = Ap[x']

4) A obtiene x' descifrando e':

   x' - As[e'], o x' = As[Ap[x']]

Por este medio, A y B han intercambiado la información secreta x y x'. Esta información no puede ser obtenida por ninguno otro que A y B, siempre que sus claves secretas no sean reveladas.

Un intercambio tal puede servir para verificar sus identidades, así como para transferir la información secreta entre las partes. Específicamente, A y B se identifican por su posesión de las claves secretas de descifrado, As y Bs respectivamente. A puede determinar si B está en posesión de la clave secreta de descifrado, Bs, haciendo retornar parte de su información x en el mensaje x' de B. Esto le indica a A que la comunicación está teniendo lugar con el propietario de Bs. B puede, de manera similar, probar la identidad de A.

Es una propiedad de algunos CSCP que los pasos de descifrado y cifrado puedan invertirse, como en D = Xp[Xs[D]]. Esto permite que una información que pudiera haber sido originada solamente por X, sea legible por cualquier usuario (que esté en posesión de Xp). Esto puede usarse por consiguiente al certificar la fuente de información, y es la base para las firmas digitales. Solamente los CSCP que tienen esta propiedad (permeabilidad) son apropiados para uso en este marco de autenticación. En el anexo C se describe uno de estos algoritmos.

Para más información, véase:

DIFFIE, W. y HELLMAN, M. E. (Noviembre 1976) - New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22, N.° 6.

ANEXO C

(a la Recomendación X.509)

## El criptosistema de claves públicas RSA

Este anexo no forma parte integrante de esta Recomendación.

*Nota* - El criptosistema especificado en este anexo fue creado por R. L. Rivest, A. Shamir y L. Adleman, y se conoce generalmente por algoritmo RSA.

### C.1 *Alcance y campo de aplicación*

Está fuera del alcance de este anexo discutir el RSA en su totalidad. Sin embargo, se da una breve descripción sobre el método, el cual se basa en el uso de exponenciación modular.

### C.2 *Referencias*

Para más información, véase:

1) Aspectos generales

RIVEST, R. L., SHAMIR, A. y ADLEMAN, L. (Febrero 1978) - A Method for Obtaining Digital Signatures and Public-key Cryptosystems, *Communications of the ACM*, 21, 2, 120-126.

2) Generación de claves

GORDON, J. - Strong RSA Keys, *Electronics Letters*, 20, 5, 514-516.

3) Descifrado

QUISQUATER, J. J., y COUVREUR, C. (14 octubre 1982) - Fast Decipherment Algorithm for RSA Public-key Cryptosystems, *Electronics Letters*, 18, 21, 905-907.

### C.3 *Definiciones*

a) *Clave pública:* el par de parámetros formado por el exponente público y el módulo aritmético.

*Nota 1* - El elemento de datos NSA.1 **subjectPublicKey**, definido como **BIT STRING** (véase el anexo G) debe interpretarse en el caso de los sistemas RSA como si fuese del tipo:

**SEQUENCE {INTEGER, INTEGER}**

donde el primer entero es el módulo aritmético y el segundo es el exponente público. La secuencia se representa por medio de las reglas de codificación básicas de NSA.1.

b) *Clave secreta:* el par de parámetros formado por el exponente secreto y el módulo aritmético.

### C.4 *Símbolos y abreviaturas*

X,Y bloques de datos que son aritméticamente menores que el módulo

n el módulo aritmético

e el exponente público

d el exponente secreto

p,q los números primos cuyo producto forma el módulo aritmético (n)

*Nota* - Se prefiere utilizar dos números primos; sin embargo, no se excluye el uso de un módulo con tres o más factores primos.

mod n módulo aritmético n.

C.5 *Descripción*

Este algoritmo asimétrico usa la función de potenciación para la transformación de bloques de datos tales que:

$y = X^e \bmod n$ con $0 \leq X < n$

$X = Y^d \bmod n$ $\quad 0 \leq Y < n$

que puede ser satisfecha, por ejemplo, por

$ed \bmod \text{lcm}\,(p-1,q-1) = 1$, o

$ed \bmod (p-1)(q-1) = 1$

Para efectuar este proceso, un bloque de datos debe interpretarse como un entero. Esto se obtiene considerando que el bloque completo de datos es una secuencia ordenada de bits (por ejemplo, de longitud l). El entero se forma entonces como la suma de los bits después de darle un peso de $2^{l-1}$ al primer bit, y dividiendo el peso por 2 para cada bit ulterior (el último bit tiene un peso de 1).

La longitud del bloque de datos debe ser el mayor número de octetos que contienen menos bits que el módulo. Los bloques incompletos deben ser rellenados de cualquier manera deseada. Se puede añadir cualquier número de bloques de relleno adicionales.

C.6 *Requisitos de seguridad*

C.6.1 *Longitudes de las claves*

Se reconoce que la longitud aceptable de la clave es probable que cambie con el tiempo, en función del costo y la disponibilidad del soporte físico, el tiempo necesario, los avances en las técnicas y el nivel de seguridad requerido. Se recomienda adoptar inicialmente para la longitud de n un valor de 512 bits, pero sujeto a *estudio ulterior*.

C.6.2 *Generación de claves*

La seguridad del criptosistema RSA se basa en la dificultad de factorizar n. Hay muchos algoritmos para realizar esta operación, y para obstaculizar el uso de cualquier técnica actualmente conocida, los valores p y q tienen que ser escogidos cuidadosamente, de acuerdo a las reglas siguientes [por ejemplo, véase la referencia 2), § C.2]:

a) deben ser escogidos al azar;

b) deben ser grandes;

c) deben ser números primos;

d) $|p-q|$ debe ser grande;

e) $(p+1)$ tendrá un factor primo grande;

f) $(q+1)$ tendrá un factor primo grande;

g) $(p-1)$ tendrá un factor primo grande, por ejemplo, r;

h) $(q-1)$ tendrá un factor primo grande, por ejemplo, s;

i) $(r-1)$ tendrá un factor primo grande;

j) $(s-1)$ tendrá un factor primo grande.

Después de generar las claves pública y secreta "Xp" y "Xs" constituidos por d, e y n, los valores p y q junto con todos los otros datos producidos tales como el producto $(p-1)(q-1)$ y los factores primos grandes deben ser preferiblemente destruidos. Sin embargo, el mantener p y q localmente puede mejorar el rendimiento en la descripción por un factor de uno a cuatro. La decisión de mantener p y q se considera un asunto local [referencia 3)].

Se tiene que asegurar que $e > \log_2(n)$ para prevenir el ataque tomando la ésima raíz mod n para revelar el texto sencillo.

C.7 *Exponente público*

El exponente público (e) debe ser común al entorno total, para minimizar la longitud de esa parte de la clave pública que, en efecto, tiene que ser distribuida, para reducir la capacidad de transmisión y la complejidad de la transformación (véase la nota 1).

El exponente e debe ser suficientemente grande, pero hasta un punto tal que la exponenciación pueda ser realizada eficientemente con respecto al tiempo de procesamiento y a la capacidad de almacenamiento. Por consiguiente se recomienda que el exponente e sea el número Fermat $F_4$ (véase la nota 2).

$F_4 = 2^{2^4} + 1$

= 65537 decimal, y

= 1 0000 0000 0000 0001 binario.

*Nota 1* - Aunque el módulo n y el exponente e son públicos, el módulo no debe ser la parte que es común a un grupo de usuarios. El conocimiento del módulo "n", del exponente público "e" y del exponente secreto "d" es suficiente para determinar la factorización de "n". Por tanto, si el módulo fuera común, todo el mundo podría deducir sus factores, y todo el mundo podría averiguar el exponente secreto de todos los demás.

*Nota 2* - El exponente fijo tiene que ser grande y primo pero también tiene que permitir un procesamiento eficiente. El número Fermat $F_4$ cumple estos requisitos, por ejemplo, la autenticación necesita solamente 17 multiplicaciones y es en promedio 30 veces más rápida que el descifrado.

### C.8 *Conformidad*

Aunque este anexo especifica un algoritmo para las funciones pública y secreta, no define el método para efectuar los cálculos; por consiguiente, pueden existir distintos productos conformes con este anexo y que sean mutuamente compatibles.

## ANEXO D

(a la Recomendación X.509)

## Funciones hash

Este anexo no forma parte integrante de esta Recomendación.

### D.1 *Requisitos de las funciones hash*

Para poder usar una función hash como una función unidireccional segura es condición indispensable que no sea posible obtener fácilmente el mismo resultado hash a partir de diferentes combinaciones del mensaje de entrada.

Una función hash fuerte cumplirá los siguientes requisitos:

a) La función hash tiene que ser unidireccional, es decir, dado un resultado hash posible cualquiera, tiene que ser computacionalmente imposible construir un mensaje de entrada que dé como hash este resultado.

b) La función hash tiene que estar libre de colisiones, es decir, tiene que ser computacionalmente imposible construir dos mensajes de entrada distintos que den en hash este mismo resultado.

### D.2 *Descripción de una función hash*

La siguiente función hash ("cuadrado-mod-n") realiza la compresión de los datos en un bloque, operando bloque por bloque.

El hasheado se hace en tres pasos principales:

1) La cadena de datos que se va a hashear se divide en bloques B de la misma longitud. Esta longitud se determina por las características del criptosistema asimétrico que se usa para el firmado. Con el criptosistema RSA, esta longitud (en octetos) es el mayor valor entero de l, para el que en módulo n se cumple:

   $16\ l < \log_2 n$.

2) Por razones de no-invertibilidad, cada octeto del bloque se divide por la mitad. Cada una de las mitades es hasheada 'rellenada' por unos binarios. Por medio de esta zonificación, se introduce una rigidez o redundancia que incrementa considerablemente la propiedad de no invertibilidad de la función hash. Cada bloque generado en el paso 1 se ensancha a la longitud del módulo n.

3) Cada bloque resultante del paso 2 se suma en módulo 2 al bloque precedente, se eleva al cuadrado y se reduce en módulo n, hasta que todos los m bloques son procesados m.
sigue:

Así pues, el resultado es el valor $H_m$, donde:

$H_0 = 0$
$H_i = (H_{i-1} \oplus B_i)^2 \text{mod } n$, para $1 \leq i \leq m$

Si el último bloque está incompleto, se rellena con "1"s.

## ANEXO E

(a la Recomendación X.509)

### Peligros contra los que ofrece protección el método de autenticación fuerte

Este anexo no forma parte integrante de esta Recomendación.

El método de autenticación fuerte que se describe en esta Recomendación ofrece protección contra los peligros como se describe en el anexo A para la autenticación fuerte.

Además, hay una gama de peligros potenciales que son específicos del propio método de autenticación fuerte. Estos peligros son:

*Comprometer la clave secreta del usuario* - uno de los principios básicos de autenticación fuerte es que la clave secreta del usuario permanezca segura. Un número de métodos prácticos están disponibles para que el usuario mantenga su clave secreta en una forma que ofrezca la seguridad adecuada. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene ese usuario.

*Comprometer la clave secreta de la AC* - el hecho de que la clave secreta de una AC permanezca segura es también un principio básico de la autenticación fuerte. La seguridad física y los métodos "necesidad de conocer" se aplican. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene cualquier usuario certificado por esa AC.

*Inducir a error a la AC para que cree un certificado no válido* - el hecho de que las AC funcionen "fuera de línea" da cierta protección. Recae sobre la AC el trabajo de comprobar que las credenciales fuertes contempladas son válidas, antes de crear un certificado. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene el usuario para el cual se creó el certificado, y cualquiera afectado por el certificado no válido.

*Colusión entre una AC deshonesta y un usuario* - un ataque de este tipo hará fracasar este método. Esto podría constituir una traición a la confianza depositada en la AC. Las consecuencias de una AC deshonesta se limitan a un trastorno de la comunicación en que interviene cualquier usuario certificado por esa AC.

*Falsificación de un certificado* - el método de autenticación fuerte protege contra la falsificación de un certificado consiguiendo que lo firme la AC. El método depende del mantenimiento del secreto de la clave secreta de la AC.

*Falsificación de un testigo* (token) - el método de autenticación fuerte protege contra la falsificación consiguiendo que lo firme el emisor. El método depende del mantenimiento del secreto de la clave secreta del emisor.

*Reactuación de un testigo* - los métodos de autenticación unidireccionales y bidireccionales protegen contra la reactuación de un testigo por medio de la inclusión de una indicación de tiempo en el testigo. El método tridireccional lo hace por medio de la comprobación de los números aleatorios.

*Ataque al sistema criptográfico* - los adelantos conseguidos en la teoría de los números, basados en las nuevas técnicas computacionales se reflejan en una mayor probabilidad de eficaces criptoanálisis de los sistemas; de ahí que sea razonable pensar en claves de mayor longitud.

## ANEXO F

(a la Recomendación X.509)

### Confidencialidad de los datos

Este anexo no forma parte integrante de esta Recomendación.

F.1 *Introducción*

El proceso de confidencialidad de los datos puede iniciarse después de que las claves necesarias para el cifrado hayan sido intercambiadas. Esto pudiera efectuarse por un intercambio previo de autenticación tal como se describe en el § 9 o por algún otro proceso de intercambio de claves; esto último está fuera del alcance de este anexo.

La confidencialidad de los datos puede ofrecerse ya sea por la aplicación de un esquema de cifrado asimétrico o por un esquema de cifrado simétrico.

F.2 *Confidencialidad de los datos por cifrado asimétrico*

En este caso la confidencialidad de los datos se obtiene cuando un originador cifra los datos que va a enviar usando la clave pública del receptor previsto: el receptor los descifrará usando su clave secreta.

F.3 *Confidencialidad de los datos por cifrado simétrico*

En este caso la confidencialidad de los datos se logra mediante un algoritmo de cifrado simétrico. Su selección está fuera del ámbito del marco de autenticación.

Cuando un intercambio de autenticación de acuerdo al § 9 se ha llevado a cabo por las dos partes interesadas, se puede derivar una clave para el uso de un algoritmo simétrico. La selección de claves secretas depende de la transformación que se utilice. Las partes tienen que estar seguras de que son claves fuertes. Esta Recomendación no especifica cómo se hace esta selección, aunque es evidente que esto debería ser acordado por las partes interesadas, o especificado en otras Recomendaciones.

## ANEXO G

(a la Recomendación X.509)

### Marco de autenticación en ASN.1

Este anexo forma parte de la Recomendación.

Este anexo incluye todas las definiciones de tipo, macro y valor NSA.1, contenidas en esta Recomendación, en la forma del módulo NSA.1 "**AuthenticationFramework**".

```
AuthenticationFramework {joint-iso-ccitt ds(5) modules(1)
                         authenticationFramework(7)}

DEFINITIONS ::=
BEGIN

EXPORTS AlgorithmIdentifier, AuthorityRevocationList, CACertificate, Certificate,
        Certificates, CertificationPath, CertificateRevocationList, UserCertificate,
        CrossCertificatePair, UserPassword, ALGORITHM,
        ENCRYPTED, PROTECTED, SIGNATURE, SIGNED;
```

```
IMPORTS
    informationFramework, selectedAttributeTypes, upperBounds
        FROM UsefulDefinitions {joint-iso-ccitt ds(5)modules(1)
                                usefulDefinitions(0)}
      Name, ATTRIBUTE,ATTRIBUTE-SYNTAX
        FROM InformationFramework informationFramework

ub-user-passwordFROM UpperBounds upperBounds;

-- tipos

Certificate        ::=  SIGNED SEQUENCE{
                            version                 [0] Version DEFAULT 1988,
                            serialNumber            SerialNumber,
                            signature               AlgorithmIdentifier,
                            issuer                  Name,
                            validity                Validity,
                            subject                 Name,
                            subjectPublicKeyInfo    SubjectPublicKeyInfo}

Version            ::=  INTEGER { 1988(0)}
SerialNumber       ::=  INTEGER

Validity           ::=  SEQUENCE{
                            notBefore               UTCTime
                            notAfter                UTCTime}

SubjectPublicKeyInfo ::= SEQUENCE{
                              algorithm             AlgorithmIdentifier
                              subjectPublicKey      BIT STRING}

AlgorithmIdentifier  ::= SEQUENCE{
                              algorithm             OBJECT IDENTIFIER,
                              parameters            ANY DEFINED BY algorithm OPTIONAL}

Certificates ::=         SEQUENCE{
                            certificate             Certificate,
                            certificationPath       ForwardCertificationPath OPTIONAL}

ForwardCertificationPath ::=    SEQUENCE OF CrossCertificates

CertificationPath ::=           SEQUENCE{
                                 userCertificate    Certificate,
                                 theCACertificates  SEQUENCE OF CertificatePair
                                                          OPTIONAL}

CrossCertificates  ::=  SET OF Certificate

CertificateList    ::=  SIGNED SEQUENCE{
                          signature             AlgorithmIdentifier,
                          issuer                Name,
                          lastUpdate            UTCTime,
                          revokedCertificates       SIGNEDSEQUENCE OF SEQUENCE{
                                                      signature       AlgorithmIdentifier,
                                                      issuer          Name,
                                                      userCertificate SerialNumber,
                                                      revocationDate  UTCTime}
                                                          OPTIONAL}

CertificatePair  ::= SEQUENCE{
                        forward [0]     Certificate OPTIONAL,
                        reverse [1]     Certificate OPTIONAL
                        -- por lo menos uno de los certificados del par debe estar presente --}

-- tipos de atributo

UserCertificate          ::=  ATTRIBUTE
                                  WITH ATTRIBUTE-SYNTAXCertificate

CACertificate            ::=  ATTRIBUTE
                                  WITH ATTRIBUTE-SYNTAXCertificate
```

```
CrossCertificatePair            ::=     ATTRIBUTE
                                            WITH ATTRIBUTE-SYNTAXCertificatePair

CertificateRevocationList       ::=     ATTRIBUTE
                                            WITH ATTRIBUTE-SYNTAXCertificateList

AuthorityRevocationList         ::=     ATTRIBUTE
                                            WITH ATTRIBUTE-SYNTAXCertificateList

UserPassword                    ::=     ATTRIBUTE
                                            WITH ATTRIBUTE-SYNTAX
                                                OCTETSTRING(SIZE(0...ub-user-password))
                                                MATCHES FOR EQUALITY

-- macros

ALGORITHM MACRO    ::=
BEGIN
TYPE NOTATION      ::= "PARAMETER"  type
VALUE NOTATION     ::= value(VALUE OBJECT IDENTIFIER)
END -- of ALGORITHM

ENCRYPTED MACRO    ::=
BEGIN

TYPE NOTATION   ::= type (ToBeEnciphered)
VALUENOTATION ::= value (VALUE BIT STRING
     -- el valor de la cadena de bits se genera
     -- tomando dos octetos que forman la codificación completa
     -- (utilizando las Reglas de Codificación Básicas NSA.1)
     -- del valor del tipo ToBeEnciphered y aplicando
     -- un procedimiento de cifrado a esos octetos --

END

SIGNED MACRO    ::=
BEGIN
TYPE NOTATION   ::= type (ToBeSigned)
VALUE NOTATION ::= value(VALUE

SEQUENCE{
     ToBeSigned,
     AlgorithmIdentifier, -- del algoritmo utilizado para generar la firma
     ENCRYPTED OCTET STRING
          -- donde la cadena de octetos es el resultado
          -- de aplicar la función al valor de
          -- "ToBeSigned" --}
                         )
END -- of SIGNED

SIGNATURE MACRO ::=
BEGIN

TYPE NOTATION      ::= type (OfSignature)
VALUE NOTATION     ::= value(VALUE

     SEQUENCE{
          AlgorithmIdentifier,
          -- del algoritmo utilizado para computar la firma
     )    ENCRYPTED OCTET STRING
          -- donde la cadena de octetos es una función, (por ejemplo, una versión comprimida
          o hasheada)
          -- del valor "OfSignature", que puede incluir el identificador del
          -- algoritmo utilizado para computar la firma --}
                                                  )
END -- of SIGNATURE

PROTECTED MACRO ::= SIGNATURE

END -- de Definiciones del Marco de Autenticación
```

ANEXO H

(a la Recomendación X.509)

## Definición de referencia de los identificadores de objeto para algoritmo

Este anexo no forma parte integrante de la Recomendación.

Este anexo define los identificadores de objeto asignados a los algoritmos de autenticación y encripción, en ausencia de un registro formal. Se tiene la intención de utilizar esos registros cuando estén disponibles. Las definiciones se presentan en forma del módulo NSA.1, **AlgorithmObjectIdentifiers**.

```
AlgorithmObjectIdentifiers      {joint-iso-ccitt ds(5) modules(1)
                                    algorithmObjectIdentifiers(8)}

DEFINITIONS ::=
BEGIN

EXPORTS
    encryptionAlgorithm, hashAlgorithm, signatureAlgorithm,
    rsa,squareMod-n,sqMod-nWithRSA;

IMPORTS
    algorithm,authenticationFramework
        FROM UsefulDefinitions  {joint-iso-ccitt ds(5)modules(1)
                                 usefulDefinitions(0)}

    ALGORITHM  FROM AuthenticationFramework authenticationFramework;

-- categorías de identificador de objeto

encryptionAlgorithm OBJECT IDENTIFIER     ::=  {algorithm 1}

hashAlgorithm OBJECT IDENTIFIER           ::=  {algorithm 2}

signatureAlgorithm OBJECT IDENTIFIER      ::=  {algorithm 3}

-- algoritmos

rsa  ALGORITHM
     PARAMETER KeySize
     ::=  {encryptionAlgorithm 1}

KeySize ::=      INTEGER

sqMod-n ALGORITHM
     PARAMETER BlockSize
     ::=  {hashAlgorithm 1}

BlockSize ::=    INTEGER

sqMod-nWithRSA ALGORITHM
     PARAMETER KeyAndBlockSize
     ::=  {signatureAlgorithm 1}

KeyAndBlockSize ::= INTEGER

END -- de definiciones de identificadores de objeto para algoritmos
```

Recommendation X.511

# THE DIRECTORY - ABSTRACT SERVICE DEFINITION [1)]

*(Melbourne, 1988)*

CONTENTS



---

1) Recommendation X.511 and ISO 9594-3, Information Processing Systems - Open Systems Interconnection - The Directory - Abstract Service Definition, were developed in close collaboration and are technically aligned.

## 0 Introducción

0.1 Este documento, junto con los otros de la serie, ha sido producido para facilitar la interconexión de los sistemas de procesamiento de información para suministrar servicios de guía. El conjunto de tales sistemas, junto con la información de guía que contienen, puede ser visto como un todo integrado, llamado la *guía*. La información contenida por la guía, conocida en su conjunto como la base de información de guía (BIG), se usa típicamente para facilitar la comunicación entre, con o sobre, objetos tales como entidades de aplicación, personas, terminales y listas de distribución.

0.2 La guía desempeña un papel importante en la interconexión de sistemas abiertos, cuyo objetivo es permitir, con un mínimo de concordancia técnica fuera de las normas de interconexión en sí, la interconexión de los sistemas de procesamiento de información:

- de diferentes fabricantes;
- sometidos a diferentes gestiones;
- de diferentes grados de complejidad; y
- de diferentes fechas de construcción.

0.3 Esta Recomendación define las capacidades suministradas por la guía a sus usuarios.

0.4 El anexo A presenta el módulo NSA.1 que contiene todas las definiciones relacionadas con el servicio abstracto.

## 1 Alcance y campo de aplicación

1.1 Esta Recomendación define de modo abstracto el servicio externamente visible suministrado por la guía.

1.2 Esta Recomendación no especifica implementaciones o productos individuales.

SECCION 1 - *Generalidades*

## 2 Referencias

Recomendación X.200 - Modelo de referencia de interconexión de sistemas abiertos para aplicaciones del CCITT

Recomendación X.208 - Especificación de la notación de sintaxis abstracta uno (NSA.1)

Recomendación X.500 - La guía - Visión de conjunto de conceptos, modelos y servicios

Recomendación X.501 - La guía - Modelos

Recomendación X.518 - La guía - Procedimientos para operación distribuida

Recomendación X.519 - La guía - Especificaciones de protocolo

Recomendación X.520 - La guía - Tipos de atributo seleccionados

Recomendación X.521 - La guía - Clases de objeto seleccionadas

Recomendación X.509 - La guía - Marco de autenticación

Recomendación X.219 - Operaciones distantes: Modelo, notación y definición del servicio

Recomendación X.229 - Operaciones a distancia: Especificación de protocolo

Recomendación X.407 - Sistemas de tratamiento de mensajes - Convenios para la definición del servicio abstracto

## 3 Definiciones

### 3.1 *Definiciones básicas de la guía*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.500:

a) *guía;*

b) *base de información de la guía (BIG);*

c) *usuario (de la guía).*

### 3.2 *Definiciones relativas al modelo de guía*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.501:

a) *agente de sistema de guía;*

b) *agente de usuario de guía.*

### 3.3 *Definiciones relativas a la base de información de la guía*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.501:

a) *asiento de alias;*

b) *árbol de información de la guía;*

c) *asiento (de guía);*

d) *superior inmediato;*

e) *asiento/objeto de inmediatamente superior;*

f) *objeto;*

g) *clase de objeto;*

h) *asiento de objeto;*

i) *subordinado;*

j) *superior.*

### 3.4 *Definiciones relativas a los asientos de la guía*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.501:

a) *atributo;*

b) *tipo de atributo;*

c) *valor de atributo;*

d) *aserción de valor de atributo.*

### 3.5 *Definiciones relativas a los nombres*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.501:

a) *alias, nombre con alias;*

b) *nombre distinguido;*

c) *nombre (de la guía);*

d) *nombre contemplado;*

e) *nombre distinguido relativo.*

### 3.6 *Definiciones relativas a las operaciones distribuidas*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.518:

a) *concatenación;*

b) *reenvío.*

3.7 *Definiciones relativas al servicio abstracto*

Esta Recomendación define los siguientes términos:

a) *filtro:* una aserción sobre la presencia o el valor de ciertos atributos de un asiento, con el fin de limitar la amplitud de una búsqueda;

b) *controles de servicio:* parámetros transportados como parte de una operación abstracta, que restringen diversos aspectos de su funcionamiento.

c) *originador:* usuario que originó una operación.

## 4 Abreviaturas

En esta Recomendación se utilizan las siguientes abreviaturas:

AIG Arbol de información de la guía

ASG Agente de sistema de guía

AVA Aserción de valor de atributo

BIG Base de información de la guía

DGG Dominio de gestión de la guía

NDR Nombre distinguido relativo

UAG Agente de usuario de guía

## 5 Convenios

En esta Recomendación se utilizan los convenios de definición de servicio abstracto definidos en la Recomendación X.407.

SECCION 2 - *Servicio abstracto*

## 6 Visión de conjunto del servicio de guía

6.1 Como se describió en la Recomendación X.501, los servicios de la guía se suministran por medio de puntos de acceso a los AUG, cada uno de los cuales actúa a nombre de un usuario. Estos conceptos se describen en la figura 1/X.511.

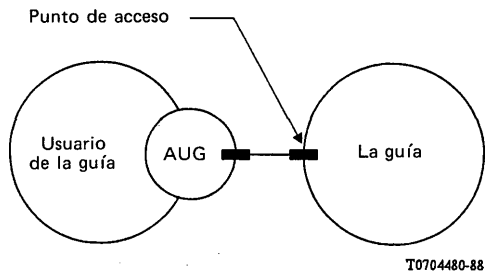



FIGURA 1/X.511

**Acceso a la guía**

6.2 En principio, los puntos de acceso a la guía pueden ser de distintos tipos, que suministran diferentes combinaciones de servicios. Conviene considerar la guía como un *objeto*, que soporta un número de tipos de *puerto*. Cada puerto define una clase particular de interacción en la cual la guía puede participar con un AUG. Cada punto de acceso corresponde a una combinación particular de tipos de puerto.

6.3 Usando la notación definida en la Recomendación X.407, la guía puede definirse como sigue:

```
directory
    OBJECT
        PORTS { readPort [S],
                searchPort [S],
                modifyPort [S]}
::= id-ot-directory
```

La guía suministra operaciones vía: Puerto de Lectura, los cuales soportan la lectura de información desde un asiento nombrado en particular en la BIG; Puertos de Búsqueda, los cuales permiten más "exploración" de la BIG; y Puertos de Modificación, que permiten la modificación de asientos en la BIG.

*Nota* - Se prevé que en el futuro podrá haber otros tipos de puerto de la guía.

6.4 Asimismo, un AUG (desde el punto de vista de la guía) puede definirse como sigue:

```
dua
    OBJECT
        PORTS { readPort [C],
                searchPort [C],
                modifyPort [C]}
::= id-ot-dua
```

El AUG consume los servicios suministrados por la guía.

6.5 Los puertos citados en los § 6.2 a 6.4 pueden definirse como sigue:

```
readPort
    PORT
        CONSUMER INVOKES {
            Read, Compare, Abandon}
::= id-pt-search

searchPort
    PORT
        CONSUMER INVOKES {
            List, Search}
::= id-pt-search

modifyPort
    PORT
    CONSUMER INVOKES {
        AddEntry, RemoveEntry,
        ModifyEntry, ModifyRDN}
    ::= id-pt-modify
```

6.6 Las operaciones del **readPort**, del **searchPort** y del **modifyPort** se definen en los § 9, 10 y 11 respectivamente.

6.7 Estos puertos sólo se utilizan como un método para estructurar la descripción del servicio de guía. La conformidad con las operaciones de la guía se especifican en la Recomendación X.519.

## 7 Tipos de información

### 7.1 *Introducción*

7.1.1 Esta cláusula identifica, y en algunos casos define, un número de tipos de información que se usan posteriormente en la definición de varias de las operaciones de la guía. Los tipos de información relacionados son aquellos que son comunes a más de una operación, que tienen probabilidades de serlo en el futuro o que son lo suficientemente complejos o autónomos como para merecer el ser definidos separadamente de la operación que los usa.

7.1.2 Varios de los tipos de información que se usan en la definición del servicio de guía se definen de hecho en otra parte. El § 7.2 identifica estos tipos e indica la fuente de su definición. Cada una de las subcláusulas restantes (§ 7.3 a 7.10) identifica y define un tipo de información.

7.2 *Tipos de información definidos en otra parte*

7.2.1 Los siguientes tipos de información se definen en la Recomendación X.501:

a) **Attribute;**

b) **AttributeType;**

c) **AttributeValue;**

d) **AttributeValueAssertion;**

e) **DistinguishedName;**

f) **Name;**

g) **RelativeDistinguishedName.**

7.2.2 El siguiente tipo de información se define en la Recomendación X.520:

a) **PresentationAddress.**

7.2.3 Los siguientes tipos de información se definen en la Recomendación X.509:

a) **Certificate;**

b) **SIGNED;**

c) **CertificationPath.**

7.2.4 El siguiente tipo de información se define en la Recomendación X.219:

a) **InvokeID.**

7.2.5 En la Recomendación X.518 se definen los siguientes tipos de información:

a) **OperationProgress;**

b) **ContinuationReference.**

7.3 *Argumentos Comunes*

7.3.1 La información de **CommonArguments** puede estar presente para calificar la invocación de cada operación que la guía puede realizar.

```
CommonArguments ::= SET {
        [30] ServiceControls DEFAULT { },
        [29] SecurityParameters DEFAULT { },
        requestor [28] DistinguishedName
                                        OPTIONAL,
        [27] OperationProgress DEFAULT notStarted,
        aliasedRDNs [26] INTEGER OPTIONAL,
        extensions [25] SET OF EXTENSION OPTIONAL}

Extension ::= SET {
        identifier      [0] INTEGER,
        critical        [1] BOOLEAN DEFAULT FALSE,
        item            [2] ANY DEFINED BY identifier}
```

7.3.2 Los diversos componentes tienen los significados definidos en los § 7.3.2.1 a 7.3.2.4.

7.3.2.1 El componente **ServiceControls** se especifica en el § 7.5. Su ausencia se considera equivalente a la existencia de un conjunto de controles vacío.

7.3.2.2 El componente **SecurityParameters** se especifica en el § 7.9. Su ausencia se considera equivalente a la existencia de un conjunto de parámetros de seguridad vacío.

7.3.2.3 El **requestor DistinguishedName** identifica al originador de una determinada operación abstracta. Contiene el nombre del usuario que está identificado en el momento de la vinculación a la guía. Puede requerirse cuando la petición ha de ser firmada (véase el § 7.10), y deberá contener el nombre del usuario que inició la petición.

7.3.2.4 El **OperationProgress** define el papel que desempeña el ASG en la evaluación distribuida de la petición. Para una descripción más detallada, véase la Recomendación X.518.

7.3.2.5 El componente **aliasedRDNs** indica al ASG que el componente objeto de la operación fue creado desreferenciando un alias en un anterior intento de operación. El valor entero indica el número de NDR, en el objeto, que provinieron de la desreferenciación del alias. (El valor se habría fijado en la respuesta de referenciamiento de la operación precedente.)

7.3.2.6 El componente **extensions** proporciona un mecanismo para expresar ampliaciones normalizadas de la forma del argumento de una operación abstracta de la guía.

*Nota* - La forma del resultado de esa operación abstracta ampliada es idéntico al de la versión no ampliada. (No obstante, el resultado de una determinada operación abstracta ampliada puede diferir de su contrapartida no ampliada.)

Los subcomponentes se definen en los § 7.3.2.6.1 a 7.3.2.6.3.

7.3.2.6.1 El **identifier** tiene por objeto identificar una ampliación particular. Los valores de este componente sólo serán asignados por futuras versiones de esta serie de Recomendaciones.

7.3.2.6.2 El subcomponente **critical** permite al originador de la operación abstracta ampliada indicar que sólo es aceptable el funcionamiento de la forma ampliada de la operación abstracta (es decir, que la forma no ampliada no es aceptable). En este caso, la ampliación es una *critical extension.* Si la guía, o alguna parte de la misma, no puede efectuar una ampliación crítica, devuelve una indicación de **ampliación crítica no disponible** (como un **error de servicio** o **calificador de resultado parcial**). Si la guía no puede realizar una ampliación que no es crítica, ignorará la presencia de la ampliación.

7.3.2.6.3 El subcomponente **ítem** proporciona la información necesaria para que la guía efectúe la operación abstracta en forma ampliada.

## 7.4 *Resultados comunes*

7.4.1 La información **CommonResults** puede estar presente para calificar el resultado de cada operación de extracción (o recuperación) que la guía puede realizar.

```
CommonResults ::= SET {
        [30] SecurityParameters        OPTIONAL,
        performer [29] DistinguishedName
                                       OPTIONAL,
        aliasDereferenced [28]         BOOLEAN
                        DEFAULT FALSE }
```

7.4.2 Los diversos componentes tienen los significados definidos en los § 7.4.2.1 a 7.4.2.3.

7.4.2.1 El componente **SecurityParameters** se especifica en el § 7.9. Su ausencia se considera equivalente a la existencia de un conjunto de parámetros de seguridad vacío.

7.4.2.2 El **performer DistinguishedName** identifica al realizador de una operación en particular. Puede requerirse cuando el resultado ha de ser firmado (véase el § 7.10), y contendrá el ASG que firmó el resultado.

7.4.2.3 El componente **alias Dereferenced** se fija a **TRUE** cuando el nombre contemplado de un objeto o base de objeto, que es la meta de la operación incluida en un alias, ha sido desreferenciado.

## 7.5 *Controles de Servicio*

7.5.1 Un parámetro **ServiceControls** contiene los controles si es que los hay, destinados a dirigir o restringir la provisión del servicio.

```
ServiceControls      ::=   SET {
    options [0] BIT STRING {
        preferChaining(0)
        chainingProhibited (1),
        localScope (2),
        dontUseCopy (3),
        dontDereferenceAliases(4)}
        DEFAULT {},
priority [1] INTEGER {
    low (0),
    medium (1),
    high (2) } DEFAULT medium,
```

**timeLimit [2]** **INTEGER OPTIONAL,**

**sizeLimit [3]** **INTEGER OPTIONAL,**

**scopeOfReferral [4] INTEGER {**
**dmd(0),**
**country(1)}**
**OPTIONAL }.**

7.5.2 Los diversos componentes tienen los significados definidos en los § 7.5.2.1 a 7.5.2.5.

7.5.2.1 El componente **options** contiene cierto número de indicaciones, cada una de las cuales, si está fijada, representa el cumplimiento de la condición sugerida. De este modo:

a) **preferChaining** indica que para prestar el servicio se debe preferir la concatenación al referimiento. La guía no está obligada a observar esa regla de preferencia.

b) **chainingProhibited** indica que esa concatenación, y otros métodos de distribuir la petición en la guía, están prohibidos.

c) **localScope** indica que la operación debe limitarse a un alcance local. La definición de esta opción es un asunto local. Por ejemplo, debe estar circunscrita a un solo ASG o a un solo DGG.

d) **dontUseCopy** indica que la información copiada (definida en la Recomendación X.518) no se utilizará para proporcionar el servicio.

e) **dontDereferenceAliases** indica que un alias utilizado para identificar el asiento afectado por una operación no ha de ser desreferenciado.

*Nota* - Esto es necesario para permitir la referencia a un asiento de alias en sí, más bien que al asiento con el alias, por ejemplo para leer el asiento de alias.

Si se omite este componente, se supone lo siguiente: no hay preferencia por la concatenación, pero ésta no está prohibida, no hay límite al alcance de la operación, el uso de copia está permitido, y los alias serán desreferenciados (salvo en las operaciones de modificación, en las cuales nunca lo son).

7.5.2.2 La **prioridad** (baja, media o alta) con la cual el servicio será suministrado. Obsérvese que este no es un servicio garantizado en el sentido de que la guía, como un todo, no implementa el hacer cola. El uso de "prioridades" en capas subyacentes no implica ninguna relación.

7.5.2.3 El **timeLimit** indica el tiempo máximo transcurrido en segundos, dentro del cual se suministrará el servicio. Si el límite no se puede cumplir, se informa un error. La ausencia de este componente significa que no hay límite de tiempo. En el caso del rebasamiento del límite de tiempo en una **lista** o **búsqueda**, el resultado es una selección arbitraria de los resultados acumulados.

*Nota* - Este componente no implica el periodo de tiempo empleado en procesar la petición, dentro del tiempo transcurrido: cualquier número de ASG puede intervenir en el procesamiento de la petición durante el tiempo transcurrido.

7.5.2.4 El **sizeLimit** sólo se aplica a las operaciones de **listado** y **búsqueda**. Indica el número máximo de objetos a devolver. Si se rebasa el límite de tamaño, los resultados del **listado** y la **búsqueda** pueden ser una selección arbitraria de los resultados acumulados, igual en número al límite de tamaño. Se descartarán todos los demás resultados.

7.5.2.5 El **scopeOfReferral** indica el alcance que tendrá un referimiento devuelto por un ASG. Según que se seleccionen los valores **dgg** o **país** se devolverán solamente referimientos a otros ASG dentro del alcance seleccionado.

Esto se aplica a los reenvíos que se hacen en el parámetro **ReferralError** y en el parámetro **no explorado** de los resultados de la **lista** y de la **búsqueda**.

7.5.3 Ciertas combinaciones de **prioridad**, **timeLimit** y **sizeLimit** pueden dar lugar a conflictos. Por ejemplo, un límite de tiempo pequeño podría entrar en conflicto con una baja prioridad; un límite de tamaño elevado podría entrar en conflicto con un límite de tiempo pequeño, etc.

7.6 *Selección de información de asiento*

7.6.1 Un parámetro **EntryInformationSelection** indica qué información está siendo solicitada de un asiento en un servicio de extracción.

```
EntryInformationSelection ::=    SET {
    attributeTypes
        CHOICE {
                allAttributes [0] NULL,
                select [1]] SET OF AttributeType
                -- conjunto vacío implica que no se
                -- solicitaron atributos --}
                DEFAULT allAttributes NULL,
    InfoTypes [2] INTEGER {
        attributeTypesOnly (0),
        attributeTypesAndValues (1) }
        DEFAULT attributeTypesAndValues }
```

7.6.2 Los diversos componentes tienen los significados definidos en los § 7.6.2.1 a 7.6.2.2.

7.6.2.1 El componente **attributeTypes** especifica el conjunto de atributos con relación a los cuales se solicitó información:

a) si se ha elegido la opción **select**, los atributos que intervienen son listados. Si la lista está vacía, no se devolverá ningún atributo. Se retornará información sobre un atributo seleccionado si el atributo está presente. Un **error de atributo** en que esté presente el problema **noSuchAttribute** no se retornará, a menos que ninguno de los atributos seleccionados esté presente;

b) si se ha seleccionado la opción **todos los atributos**, se solicitará información sobre todos los atributos en el asiento.

La información de atributo se devolverá solamente si los derechos de acceso son suficientes. Un **securityError** (con un problema de **insufficientAccessRights**) sólo se retornará cuando los derechos de acceso impidan la lectura de todos los valores de atributo solicitados.

7.6.2.2 El componente **infoTypes** especifica si se solicitan ambas informaciones, de tipo de atributo y de valor de atributo (el valor por defecto) o si se solicita solamente la información de tipo de atributo. Si el componente **attributeTypes** (§ 7.6.2.1) es tal que no solicita atributos, este componente no es significativo.

## 7.7 *Información de asiento*

7.7.1 Un parámetro **EntryInformation** transporta información seleccionada desde un asiento.

```
EntryInformation ::= SEQUENCE {
    DistinguishedName,
    fromEntry BOOLEAN DEFAULT TRUE,
    SET OF CHOICE {
        AttributeType,
        Attribute} OPTIONAL }
```

7.7.2 El **DistinguishedName** del asiento se incluye siempre.

7.7.3 El parámetro **fromEntry** indica si la información obtenida provino del asiento (**TRUE**) o de una copia del asiento (**FALSE**).

7.7.4 Se incluye un conjunto de **AttributeTypes** o **Attributes**, si procede, cada uno de los cuales puede estar solo o ir acompañado de uno o más valores de atributo.

## 7.8 *Filtro*

7.8.1 Un parámetro **Filter** aplica una prueba a un asiento particular y será o no satisfecho por el asiento. El filtro se expresa en términos de aserciones sobre la presencia o valor de ciertos atributos del asiento, y se satisface únicamente si se evalúa como **TRUE**.

*Nota* - Un filtro puede ser **TRUE**, **FALSE** o indefinido.

```
Filter      ::=   CHOICE {
    item          [0] FilterItem,
    and           [1] SET OF Filter,
    or            [2] SET OF Filter,
    not           [3] Filter }
```

```
FilterItem      ::=  CHOICE {
    equality          [0]  AttributeValueAssertion,
    substrings        [1]  SEQUENCE {
        type                   AttributeType,
        strings SEQUENCE OF CHOICE {
                       Initial    [0]  AttributeValue,
                       any        [1]  AttributeValue,
                       final      [2]  AttributeValue}},
    greaterOrEqual         [2]  AttributeValueAssertion,
    lessOrEqual            [3]  AttributeValueAssertion,
    present                [4]  AttributeType,
    approximateMatch       [5]  AttributeValueAssertion }
```

7.8.2 Un **filtro** es o bien un **FilterItem** (ver el § 7.8.3) o una expresión que comprende **filtros** más sencillos reunidos usando los operadores lógicos **and**, **or** y **not**. El **filtro** es indefinido si es un **FilterItem** que es indefinido, o si involucra uno o más **filtros** más simples, todos ellos indefinidos. En los demás casos, el filtro es:

a) un **item**, es **TRUE** únicamente si el **FilterItem** correspondiente es **TRUE**;

b) un **and**, es **TRUE** a no ser que cualquiera de los **filtros** anidados sea **FALSE**;

*Nota* - Por consiguiente, si no hay **filtros** anidados los **and** se evalúan como **TRUE**.

c) un **or**, es **FALSE** a no ser que cualquiera de los **filtros** anidados sea **TRUE**;

*Nota* - Por consiguiente, si no hay **filtros** anidados el **or** se evalúa como **FALSE**.

d) un **not**, es **TRUE** únicamente si el **filtro** anidado es **FALSE**.

7.8.3 Un **FilterItem** es una aserción sobre la presencia, el valor o los valores de un atributo de un tipo particular en el asiento sometido a prueba. Cada aserción de este tipo es **TRUE** o **FALSE** o indefinido.

7.8.3.1 Cada **FilterItem** incluye un **AttributeType** que identifica el atributo particular en cuestión.

7.8.3.2 Cualquier aserción sobre el valor de tal atributo se define solamente si el **AttributeType** es conocido y el (los) **AttributeValue**(s) contemplado(s) se conforma(n) a la sintaxis del atributo definida para ese tipo de atributo.

*Nota 1* - Cuando no se satisfacen estas condiciones, el **FilterItem** es indefinido.

*Nota 2* - Las restricciones del control de acceso pueden requerir que se considere el **FilterItem** como indefinido.

7.8.3.3 Las aserciones sobre el valor de un atributo se evalúan usando las reglas de concordancia asociadas con la sintaxis de atributo definida para ese tipo de atributo. Una regla de concordancia no definida para una sintaxis de atributo particular no puede usarse para hacer aserciones sobre ese atributo.

*Nota* - Cuando no se satisface esta condición, el **FilterItem** es indefinido.

7.8.3.4 Un **FilterItem** puede ser indefinido (como se describe en los § 7.8.3.2 y 7.8.3.3). En los demás casos en los que el **FilterItem** determina por una aserción:

a) **equality**, es **TRUE** únicamente si hay un valor del atributo que es igual al determinado;

b) **substrings**, es **TRUE** únicamente si hay un valor del atributo en el cual las subcadenas especificadas aparecen en el orden dado. Las subcadenas tienen que ser no-superponientes y pueden (pero no tienen necesariamente que) estar separadas de los extremos del valor de atributo, y unos de otros, por cero o más elementos de cadena.

Si está presente **initial** la subcadena concordará con la subcadena inicial del valor de atributo; si está presente **final**, la subcadena concordará con la subcadena final del valor de atributo; está presente **cualquiera**, la subcadena concordará con cualquier subcadena del valor de atributo;

c) **greaterOrEqual**, es **TRUE** únicamente si la ordenación relativa (definida por el algoritmo de ordenación apropiado) coloca el valor suministrado antes que cualquier valor del atributo o lo hace igual a él;

d) **lessOrEqual**, es **TRUE** si y sólo si la ordenación relativa (definida por el algoritmo de ordenación apropiado) coloca el valor suministrado después de cualquier valor del atributo o lo hace igual a él;

e) **present**, es **TRUE** únicamente si tal atributo está presente en el asiento;

f) **approximateMatch** es **TRUE** únicamente si hay un valor de atributo que concuerda con el determinado por algún algoritmo de concordancia aproximada localmente definido (por ejemplo, variaciones ortográficas, concordancias fonéticas, etc.). En esta versión de la Recomendación no se dan orientaciones específicas sobre la concordancia aproximada. Si la concordancia aproximada no es soportada, este **ítem de filtro** debe tratarse como una concordancia por **igualdad**.

7.9 *Parámetros de seguridad*

7.9.1 Los **SecurityParameters** gobiernan la operación de varias características de seguridad asociadas con una operación de la guía.

*Nota* - Estos parámetros se transportan del expedidor al destinatario. Cuando aparecen los parámetros en el argumento de una operación abstracta, el solicitante es el expedidor y el realizador es el destinatario. En un resultado se invierten los papeles.

```
SecurityParameters ::=          SET {
    certification-path          [0]
    CertificationPath       OPTIONAL,
    name                [1]     DistinguishedName
                                    OPTIONAL,
    time                [2]     UTCTime OPTIONAL,
    random              [3]     BIT STRING OPTIONAL,
    target              [4]     ProtectionRequest  OPTIONAL
                                                   }
ProtectionRequest   ::=         INTEGER {
                                    none(0),
                                    signed (1)}
```

7.9.2 Los diversos componentes tienen los significados definidos en los § 7.9.2.1 a 7.9.2.5.

7.9.2.1 El componente **CertificationPath** consiste en el certificado del expedidor y, opcionalmente, una secuencia de pares de certificados. El certificado se utiliza para asociar la clave pública y el nombre distinguido del expedidor, y puede utilizarse para verificar la firma relativa al argumento o resultado. Este parámetro está presente y el argumento o resultado está firmado. La secuencia de pares de certificados consiste en certificados cruzados de autoridades de certificación. Se utiliza para permitir la validación del certificado del expedidor. No es necesario si el receptor tiene la misma autoridad de certificación que el expedidor. Si el receptor requiere un conjunto válido de pares de certificados, y este parámetro no está presente, la determinación de si el receptor rechaza la firma relativa al argumento o resultado o trata de generar el trayecto de certificación, es un asunto local.

7.9.2.2 El **name** es el nombre distinguido del primer receptor deseado del argumento o resultado. Por ejemplo, si un AUG genera un argumento firmado, el nombre es el nombre distinguido del ASG al cual se sometió la operación.

7.9.2.3 El **time** es el tiempo de expiración previsto para la validez de la firma, cuando se utilizan argumentos firmados. Se emplea junto con el número aleatorio para permitir la detección de intentos de maniobras falsas de reproducción.

7.9.2.4 El componente **random** es un número que debe ser diferente para cada testigo no caducado. Se utiliza junto con el parámetro de tiempo para habilitar la detección de maniobras falsas de reproducción cuando el argumento o el resultado ha sido firmado.

7.9.2.5 La **target ProtectionRequest** puede aparecer solamente en la solicitud de que se realice una operación. Indica la preferencia del solicitante en cuanto al grado de protección que se le quiere suministrar al resultado. Se prevén dos niveles: **ninguna** (no se solicita protección), y **firmado** (se solicita que la guía firme los resultados, valor por defecto). El grado de protección suministrado en efecto al resultado se indica por la forma del resultado y puede ser igual a o menor que el solicitado, lo que dependerá de las limitaciones de la guía.

7.10 *OPTIONALLY-SIGNED*

7.10.1 Un tipo de información **OPTIONALLY-SIGNED** es aquél cuyos valores pueden, a opción del generador, ser acompañados por su firma digital. Esta capacidad se especifica por medio de la siguiente macro:

```
OPTIONALLY-SIGNED MACRO ::=
BEGIN

TYPE NOTATION ::= type (Type)
VALUE NOTATION ::= value (VALUE
    CHOICE { Type, SIGNED Type})

END
```

7.10.2 La macro **SIGNED**, que describe la forma firmada de la información, se especifica en la Recomendación X.509.

## 8 Operaciones de vincular y desvincular

Las operaciones de **DirectoryBind** y **DirectoryUnbind**, definidas en los § 8.1 y 8.2 respectivamente, son usadas por el AUG al principio y al final de un periodo particular de acceso a la guía.

8.1 *Vinculación a la guía*

8.1.1 Se utiliza una operación de **DirectoryBind** al comienzo del periodo de acceso a la guía.

```
DirectoryBind ::= ABSTRACT-BIND
    TO { readPort, searchPort, modifyPort }
    BIND
    ARGUMENT DirectoryBindArgument
    RESULT DirectoryBindResult
    BIND-ERROR DirectoryBindError

DirectoryBindArgument ::= SET {
    credentials [0] Credentials OPTIONAL,
    versions [1] Versions DEFAULT
                        v1988}

Credentials ::= CHOICE {
    simple [0] SimpleCredentials,
    strong [1] StrongCredentials,
    externalProcedure [2] EXTERNAL }

SimpleCredentials ::= SEQUENCE {
    name [0] DistinguishedName,
    validity [1] SET {
        time1 [0] UTCTime OPTIONAL,
        Time2 [1] UTCTime OPTIONAL,
        random1 [2] BIT STRING OPTIONAL,
        random2 [3] BIT STRING OPTIONAL } OPTIONAL,
        -- en la mayor parte de los casos los argumentos para
        -- tiempo y aleatorio son significativos en
        -- diálogos que emplean el mecanismo de contraseña
        -- protegida y basan el significado en
        -- acuerdos bilaterales

password [2] OCTET STRING OPTIONAL }
    -- el valor podría ser una
    -- contraseña no protegida o Protegido1 o Protegido2,
    -- como se especifica en la Recomendación X.509.

StrongCredentials ::= SET {
    certification-path [0] CertificationPath
                        OPTIONAL,
    bind-token [1] Token }
```

```
Token                ::=    SIGNED SEQUENCE {
    algorithm                [0]  AlgorithmIdentifier,
    name                     [1]  DistinguishedName,
    time                     [2]  UTCTime,
    random                   [3]  BIT STRING }

Versions             ::=    BIT STRING {v1988(0)}

DirectoryBindResult  ::=        DirectoryBindArgument

DirectoryBindError           ::=    SET {
    versions          [0]            Versions DEFAULT v1988,
    CHOICE {
        serviceError      [1]  ServiceProblem
        securityError     [2]  SecurityProblem
}}
```

8.1.2 Los diversos argumentos tienen los significados definidos en los § 8.1.2.1 y 8.1.2.2.

8.1.2.1 Las **Credentials** del **DirectoryBindArgument** permiten a la guía establecer la identidad del usuario. Estas credenciales pueden ser **simples** o **fuertes** (véase la Recomendación X.509), o definidas externamente (externalProcedure).

8.1.2.1.1 Las **SimpleCredentials** consisten en un nombre (que es siempre el nombre distinguido de un objeto) y (facultativamente) una contraseña. Esto da un grado limitado de seguridad. Si la contraseña está protegida como se describe en el § 5 de la Recomendación X.509, las **SimpleCredentials** incluyen el nombre, la contraseña y (facultativamente) el tiempo (hora) y/o números aleatorios que se utilizan para evitar maniobras fraudulentas de reproducción. En algunos casos, una contraseña protegida puede ser verificada por un objeto que sólo conoce dicha contraseña tras una regeneración local del argumento de protección (contraseña). En otros casos puede ser posible una comparación directa.

8.1.2.1.2 Las **StrongCredentials** consisten en un testigo de vinculación y, opcionalmente, un certificado y una secuencia de certificados recíprocos de autoridades de certificación (definida en la Recomendación X.509). Esto permite a la guía autenticar la identidad del solicitante que establece la asociación, y viceversa. Los argumentos del testigo de vincular se utilizan como sigue: **algoritmo** es el identificador del algoritmo utilizado para firmar la información, **name** es el nombre del destinatario deseado. El parámetro **time** contiene la fecha y hora de expiración del testigo. El número **aleatorio** es un número que debe ser diferente para cada testigo no caducado, y que puede ser utilizado por el receptor para detectar maniobras fraudulentas de reproducción.

8.1.2.1.3 Si se utiliza el **externalProcedure**, la semántica del esquema de autenticación que se emplea está fuera del alcance de la recomendación sobre la guía.

8.1.2.2 El argumento **Versions** del argumento del **DirectoryBindArgument** identifica las versiones del servicio en las cuales el AUG está preparado para participar. Para esta versión del protocolo, el valor debe fijarse a **v1988**(0).

8.1.2.3 La migración a futuras versiones de la guía debe facilitarse por lo siguiente:

a) los elementos de **DirectoryBindArgument** que no sean los definidos en esta Recomendación deberán ser aceptados e ignorados;

b) las opciones adicionales para bits denominados del **DirectoryBindArgument** (por ejemplo **Versions**), no definidas, deberán ser aceptadas e ignoradas.

8.1.3 Si una petición de vinculación tiene éxito, deberá retornarse un resultado. Los parámetros de resultado tienen los significados definidos en los § 8.1.3.1 a 8.1.3.2.

8.1.3.1 Las **Credentials** del **DirectoryBindResult** permiten al usuario establecer la identidad de la guía. Estas credenciales permiten transportar al AUG información que identifica el ASG (que está prestando directamente el servicio de guía). Deberán ser de la misma forma (es decir, **CHOICE**) que las suministradas por el usuario.

8.1.3.2 El parámetro **Versions** del **DirectoryBindResult** indica cuál de las versiones del servicio solicitado por el AUG será efectivamente prestada por este ASG.

8.1.4 En caso de fracaso de la petición de vinculación, se retornará un error de vinculación como se indica en los § 8.1.4.1 a 8.1.4.2.

8.1.4.1 El parámetro **Versions** del **ErrorBindDirectory** indica cuáles versiones son soportadas por este ASG.

8.1.4.2 Un **securityError** o **serviceError** se comunicará como sigue:

- **securityError** **inappropriateAuthentication**
  **invalidCredentials**

- **serviceError** **unavailable.**

8.2 *Desvinculación de la guía*

8.2.1 Una operación de **DirectoryUnbind** se usa al final de un periodo de acceso a la guía.

```
DirectoryUnbind ::= ABSTRACT-UNBIND
    FROM {readPort, searchPort, modifyPort }
```

8.2.2 La **DirectoryUnbind** no tiene argumentos.

## 9 Operaciones de Leer la guía

Hay dos operaciones "semejantes-a-lectura": **Read** y **Compare**, definidas en los § 9.1 y 9.2 respectivamente. La operación **Abandon** definida en el § 9.3, se agrupa con las operaciones de leer (Read) por razones de conveniencia.

9.1 *Leer (Read)*

9.1.1 Una operación **Read** se usa para extraer información de un asiento identificado explícitamente. También puede ser usada para verificar un nombre distinguido. Los argumentos de la operación pueden ser firmados opcionalmente (véase el § 7.10) por el solicitante. Si así se solicita, la guía puede firmar el resultado.

```
Read ::= ABSTRACT-OPERATION
    ARGUMENT ReadArgument
    RESULT ReadResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, Abandoned,
        SecurityError }

ReadArgument ::= OPTIONALLY-SIGNED SET {
    object [0] Name,
    selection [1] Selection F[13] EntryInformationSelection
        DEFAULT {}
    COMPONENTS OF CommonArguments }

ReadResult ::= OPTIONALLY-SIGNED SET {
    entry [0] EntryInformation,
    COMPONENTS OF CommonResults }
```

9.1.2 Los diversos argumentos tienen los significados definidos en los § 9.1.2.1 a 9.1.2.3.

9.1.2.1 El argumento **object** identifica el asiento objeto del cual se solicita información. Si el **Name** comprende uno o más alias, estos son desreferenciados (a no ser que esto sea prohibido por los controles del servicio pertinentes).

9.1.2.2 El argumento **selection** indica qué información del asiento se solicita (véase el § 7.6).

9.1.2.3 Los **CommonArguments** (véase el § 7.3) incluyen una especificación de los controles de servicio que se aplican a la solicitud. Para los fines de esta operación, el componente **sizeLimit** no es pertinente y no se tiene en cuenta si se suministra.

9.1.3 Si la solicitud tiene éxito, el resultado será retornado. Los parámetros de resultado tienen los significados definidos en los § 9.1.3.1 y 7.4.

9.1.3.1 El parámetro de resultado **entry** contiene la información solicitada (véase el § 7.7).

9.1.4 Si fracasa la petición, se comunica uno de los errores enumerados. Si no puede devolverse ninguno de los atributos explícitamente enumerados en **selección**, se informará un **AttributeError** con el problema **noSuchAttribute**. Las circunstancias en las cuales se informan otros errores se definen en el § 12.

9.2 *Comparar (Compare)*

9.2.1 Una operación **Compare** se usa para comparar un valor (el cual se suministra como un argumento de la solicitud) con el valor o los valores de un tipo de atributo determinado en un asiento objeto determinado. Los argumentos de la operación pueden ser firmados opcionalmente (véase el § 7.10) por el solicitante. Si así se solicita, la guía puede firmar el resultado.

```
Compare ::=          ABSTRACT-OPERATION
    ARGUMENT         CompareArgument
    RESULT           CompareResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, Abandoned,
        SecurityError }

CompareArgument ::=          OPTIONALLY-SIGNED
SET {
    object          [0]      Name,
    purported       [1]      AttributeValueAssertion,
    COMPONENTS OF CommonArguments }

CompareResult ::=            OPTIONALLY-SIGNED
SET {
    DistinguishedName OPTIONAL,
    matched         [0]      BOOLEAN,
    from Entry      [1]      BOOLEAN DEFAULT TRUE,
    COMPONENTS OF CommonResults }
```

9.2.2 Los diversos argumentos tienen los significados definidos en los § 9.2.2.1 a 9.2.2.3.

9.2.2.1 El argumento **object** es el nombre del asiento de objeto en cuestión. Si el **name** comprende uno o más alias, estos son desreferenciados (a menos que ello esté prohibido por un control de servicio aplicable).

9.2.2.2 El argumento **purported** identifica el tipo de atributo y el valor que ha de compararse con el contenido en el asiento.

9.2.2.3 Los **CommonArguments** (véase el § 7.3) especifican los controles de servicio aplicables a la petición. A los fines de esta operación, el componente **sizeLimit** no es significativo y, si se suministra, será ignorado.

9.2.3 Si la petición tiene éxito (es decir, si se efectúa realmente la comparación), se retornará el resultado. Los parámetros de resultado tienen los significados descritos en los § 9.2.3.1 a 9.2.3.3 y el § 7.4.

9.2.3.1 El **DistinguishedName** está presente si un alias fue desreferenciado y representa el nombre distinguido del propio objeto.

9.2.3.2 El parámetro de resultado **matched** contiene el resultado de la comparación. Este parámetro adopta el valor **TRUE** si los valores fueron concordados y comparados, y **FALSE** en otro caso.

9.2.3.3 Si **fromEntry** es **TRUE**, la información fue comparada con el asiento. Si es **FALSE**, alguna parte de la información fue comparada con una copia.

9.2.4 En caso de fracaso de la petición, se comunica uno de los errores enumerados. Las circunstancias en que se comunican estos errores se definen en el § 12.

9.3 *Abandono (Abandon)*

9.3.1 Las operaciones que interrogan la guía pueden ser abandonadas utilizando la operación **Abandon** si el usuario ya no está autorizado en el resultado.

```
Abandon ::=          ABSTRACT-OPERATION
    ARGUMENT         AbandonArgument
    RESULT           AbandonResult
    ERRORS {AbandonFailed}

AbandonArgument ::=  SEQUENCE {
    InvokeID        [0]  InvokeID}

AbandonResult ::=    NULL
```

9.3.2 Hay un solo argumento el **InvokeID** que identifica la operación que ha de abandonarse. El valor de **invokeID** es el mismo del **invokeID** utilizado para invocar la operación que va a ser abandonada.

9.3.3 Si la solicitud tiene éxito, se retornará un resultado, aunque no se transporte información con el mismo. La operación original fracasará con un error **abandonado**.

9.3.4 Si fracasa la solicitud, se comunica el error **AbandonFailed**. Este error se describe en el § 12.3.

9.3.5 **Abandonar** sólo es aplicable a operaciones de interrogación, es decir, **Read**, **Compare**, **List** y **Search**.

9.3.6 Un ASG puede abandonar una operación localmente. Si el ASG ha concatenado o difundido la operación a otros ASG, podrá, a su vez, pedir a estos que abandonen la operación. Un ASG podrá no abandonar la operación, en cuyo caso devolverá el error **AbandonFailed**.

## 10 Operaciones de búsqueda en guía

Hay dos operaciones "semejantes-a-búsqueda": **List** y **Search**, definidas en el § 10.1 y 10.2 respectivamente.

### 10.1 *Listar (List)*

10.1.1 Una operación **List** se usa para obtener una lista de los subordinados inmediatos de un asiento identificado explícitamente. En ciertas circunstancias, la lista retornada puede estar incompleta. Los argumentos de la operación pueden ser firmados opcionalmente (véase el § 7.10) por el solicitante. Si así se solicita, la guía puede firmar el resultado.

```
List ::= ABSTRACT-OPERATION
    ARGUMENT        ListArgument
    RESULT          ListResult
    ERRORS {
                            NameError
        ServiceError, Referral, Abandoned,
        SecurityError }

List Argument ::= OPTIONALLY-SIGNED SET {
    object    [0]  Name,
    COMPONENTS OF CommonArguments }

ListResult      ::= OPTIONALLY-SIGNED
CHOICE {
    listInfo SET {
    DistinguishedName OPTIONAL,
    subordinates    [1]  SET OF SEQUENCE {
        RelativeDistinguishedName,
        aliasEntry      [0]  BOOLEAN DEFAULT FALSE
        fromEntry       [1]  BOOLEAN DEFAULT TRUE},

    partialOutcomeQualifier    [2]
                        PartialOutcomeQualifier    OPTIONAL
    COMPONENTS OF CommonResults },
    uncorrelatedListInfo        [0]  SET OF
                                ListResult }

PartialOutcomeQualifier  ::= SET {
    limitProblem        [0]  LimitProblem
        OPTIONAL,
    unexplored          [1]  SET OF
        ContinuationReference   OPTIONAL,
    unavailableCriticalExtensions  [2]  BOOLEAN DEFAULT FALSE }

LimitProblem  ::= INTEGER {
    timeLimitExceeded (0),
    sizeLimitExceeded (1),
    administrativeLimitExceeded (2) }
```

10.1.2 Los diversos argumentos tienen los significados definidos en los § 10.1.2.1 y 7.3.

10.1.2.1 El argumento **object** identifica el asiento objeto (o posiblemente la raíz) cuyos subordinados inmediatos serán listados. Si el **Name** comprende uno o más alias, éstos son desreferenciados (a menos que sean prohibidos por el control de servicio pertinente).

10.1.3 La petición tiene éxito si el objeto es localizado con independencia de si hay cualquier información de subordinado por devolver. Los parámetros de resultado tienen los significados definidos en los § 10.1.3.1 a 10.1.3.4 y 7.4.

10.1.3.1 El **DistinguishedName** está presente si un alias fue desreferenciado. Representa el nombre distinguido del propio objeto.

10.1.3.2 El parámetro **subordinates** transporta la información sobre los subordinados inmediatos, si existen, del asiento denominado. Si alguno de los asientos subordinados son alias, no serán desreferenciados.

10.1.3.2.1 El **RelativeDistinguishedName** es el del subordinado.

10.1.3.2.2 El parámetro **fromEntry** indica si la información se obtuvo del asiento (**TRUE**) o de una copia del asiento (**FALSE**).

10.1.3.2.3 El parámetro **aliasEntry** indica si el asiento subordinado es un asiento de alias (**TRUE**) o no (**FALSE**).

10.1.3.3 El **PartialOutcomeQualifier** consta de tres subcomponentes definidos en los § 10.1.3.3.1 a 10.1.3.3.3. Este parámetro deberá estar presente cuando el resultado esté incompleto.

10.1.3.3.1 El parámetro **LimitProblem** indica que se ha rebasado el límite de tiempo, el límite de tamaño, o un límite administrativo. Los resultados devueltos son los que estaban disponibles al alcanzar el límite.

10.1.3.3.2 El parámetro **unexplored** estará presente si regiones del AIG no fueron exploradas. Su información permite al AUG continuar el procesamiento de la operación **List**, estableciendo contactos con otros puntos de acceso, si así lo desea. El parámetro consiste en un conjunto (posiblemente vacío) de **ContinuationReferences**, cada una de las cuales consta del nombre de un objeto de base a partir del cual se puede hacer avanzar la operación, un valor apropiado de **OperationProgress**, y un conjunto de puntos de acceso a partir de los cuales se puede hacer avanzar la petición. Las **ContinuationReferences** que se devuelvan estarán en el margen del reenvío solicitado en el control de servicio de operación.

10.1.3.3.3 El parámetro **unavailableCriticalExtensions** indica, cuando está presente, que una o más ampliaciones críticas no estuvieron disponibles en alguna parte de la guía.

10.1.3.4 Cuando el AUG ha solicitado la protección de una petición mediante la **firma**, el parámetro **uncorrelatedListInfo** puede comprender un número de conjuntos de parámetros de resultado provenientes de, y firmados por, diferentes componentes de la guía. Si ningún ASG en la cadena puede correlacionar todos los resultados, el AUG, para disponer de un resultado deberá reunir los diversos componentes.

10.1.4 Si falla la solicitud, uno de los errores listados será informado. Las circunstancias en las cuales los distintos errores serán informados se definen en el § 12.

## 10.2 *Buscar (Search)*

10.2.1 Una operación **Search** se usa para explorar una parte del AIG para buscar asientos de interés, y para retornar información seleccionada desde esos asientos. Los argumentos de la operación pueden ser firmados opcionalmente (véase el § 7.10) por el solicitante. Si así se solicita, la guía puede firmar el resultado.

```
Search      ::=  ABSTRACT-OPERATION
    ARGUMENT        SearchArgument
    RESULT          SearchResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, Abandoned,
        SecurityError }
SearchArgument::=  OPTIONALLY-SIGNED
SET {
    baseObject          [0]  Name,
    subset              [1]  INTEGER {
        baseObject (0),
        oneLevel(1),
        wholeSubtree(2)}     DEFAULT baseObject,
    filter              [2]  Filter DEFAULT and {}.
```

```
        searchAliases       [3]  BOOLEAN DEFAULT TRUE,
        selection           [4]  EntryInformationSelection DEFAULT {}

            COMPONENTS OF CommonArguments }

SearchResult ::=        OPTIONALLY-SIGNED
    CHOICE {
    searchInfo SET {
    DistinguishedName OPTIONAL,
    entries [0] SET OF EntryInformation,
    partialOutcomeQualifier
        [2]PartialOutcomeQualifier OPTIONAL,
COMPONENTS OF CommonResults },
uncorrelatedSearchInfo [0] SET OF
    SearchResult }
```

10.2.2 Los diversos argumentos tienen los significados definidos en los § 10.2.2.1 a 10.2.2.3, 10.2.2.5, y 7.3.

10.2.2.1 El argumento **baseObject** identifica el asiento de objeto (o posiblemente la raíz) con relación al cual se efectúa la búsqueda.

10.2.2.2 El argumento **subset** indica si la búsqueda se aplica:

a) solamente al **baseObject**;

b) solamente a los subordinados inmediatos del objeto de base (**oneLevel**);

c) al objeto de base y a todos sus subordinados (**wholeSubtree**).

10.2.2.3 El argumento **filtro** se utiliza para eliminar, del espacio de búsqueda, los asientos que no ofrecen interés. Sólo se retornará información sobre asientos qué satisfacen el filtro (véase el § 7.8).

10.2.2.4 Los alias serán desreferenciados mientras se localiza el objeto de base, atendiendo al valor fijado del **control del servicio dontDereferenceAliases.** Los alias contenidos entre los subordinados del objeto de base serán desreferenciados durante la búsqueda, con arreglo al valor del parámetro de **searchAliases.** Si el parámetro de **searchAliases** es **TRUE,** los alias serán desreferenciados; si el parámetro es **FALSE,** los alias no serán desreferenciados. Si el parámetro **searchAliases** es **TRUE** continuará la búsqueda en el subárbol del objeto designado por un seudónimo.

10.2.2.5 El argumento **selection** indica la información que se solicita entre las contenidas en los asientos (véase el § 7.6).

10.2.3 La petición tiene éxito si el objeto de base es localizado independientemente de que se devuelvan subordinados.

*Nota* - Como un corolario de esto, el resultado de una **búsqueda** (no filtrada) aplicada a un solo asiento puede no ser idéntico a una **lectura** que trata de interrogar el mismo conjunto de atributos de la entrada. Esto es así porque la lectura retornará un **error de atributo** si ninguno de los atributos seleccionados existen en el asiento.

Los parámetros de resultado tienen los significados definidos en los § 10.2.3.1 a 10.2.3.4 y 7.3.

10.2.3.1 El **DistinguishedName** está presente si un alias fue desreferenciado, y representa el nombre distinguido del objeto de base.

10.2.3.2 El parámetro **entries** transporta la información solicitada de cada asiento (cero o más) que satisface el filtro (véase el § 7.5).

10.2.3.3 El **PartialOutcomeQualifier** consta de dos subcomponentes descritos en el funcionamiento de la **lista** en el § 10.1.3.

10.2.3.4 El parámetro **uncorrelatedSearchInfo** se describe como se hizo para la **uncorrelatedListInfo** en el § 10.1.3.4.

10.2.4 Si fracasa la petición, se informa uno de los errores enumerados. Las circunstancias en las cuales se informan los distintos errores se definen en el § 12.

## 11 Operaciones de modificación de la guía

Hay cuatro operaciones para modificar la guía: **AddEntry**, **RemoveEntry**, **ModifyEntry**, y **ModifyRDN** definidos en los § 11.1 a 11.4 respectivamente.

*Nota 1* - En cada una de estas operaciones abstractas, el asiento sobre el que se desea actuar se identifica por su nombre *distinguido.*

*Nota 2* - El éxito de las operaciones **AddEntry**, **RemoveEntry** y **ModifyEntry** dependerá de la distribución física de la BIG en la guía. Se informará del fracaso con un **UpdateError** y el problema **affectsMultipleDSAs**. Véase la Recomendación X.518.

### 11.1 *Añadir Asiento (Add Entry)*

11.1.1 Se utiliza una operación de **AddEntry** para añadir un asiento constitutivo de hoja (un asiento de objeto o un asiento de alias) al Arbol de Información de la guía (AIG). Los argumentos de esta operación pueden ser firmados opcionalmente (véase el § 7.10) por el solicitante.

```
AddEntry        ::=  ABSTRACT-OPERATION
    ARGUMENT         AddEntryArgument
    RESULT           AddEntryResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, SecurityError,
        UpdateError }

AddEntryArgument ::=  OPTIONALLY-SIGNED
SET {
    object           [0]  DistinguishedName,
    entry            [1]  SET OF Attribute,
    COMPONENTS OF CommonArguments }

AddEntryResult   ::=  NULL
```

11.1.2 Los diversos argumentos tienen los significados definidos en los § 11.1.2.1 a 11.1.2.3.

11.1.2.1 El argumento **object** identifica el asiento que será añadido. El inmediato superior, que tiene que existir para que la operación tenga éxito, puede determinarse suprimiendo el último componente NDR (que pertenece al asiento a crear).

11.1.2.2 El argumento **entry** contiene la información de atributo que con la del NDR constituye el asiento a crear. La guía asegurará que el asiento concuerda con el esquema de la guía. Donde el asiento que se está creando es un alias, no se hace comprobación para asegurar que el atributo **aliasedObjectName** apunta a un asiento válido.

11.1.2.3 Los **CommonArguments** (véase el § 7.3) incluyen una especificación de los controles de servicio que se aplican a la solicitud. Para los fines de esta operación la opción **dontDereferenceAlias** y el componente **sizeLimit** no son pertinentes y son ignorados si se suministran. Los alias nunca son desreferenciados por esta operación.

11.1.3 Si la solicitud tiene éxito, un resultado será retornado, aunque no se transporte información en el mismo.

11.1.4 Si la solicitud falla, se informará uno de los errores listados. Las circunstancias en las cuales los distintos errores serán informados se definen en el § 12.

### 11.2 *Suprimir asiento (Remove Entry)*

11.2.1 Una operación **RemoveEntry** se usa para suprimir un asiento hoja (un asiento objeto o un asiento de alias) del AIG. Los argumentos de la operación pueden ser firmados opcionalmente (véase el § 7.10) por el solicitante.

```
RemoveEntry  ::=        ABSTRACT-OPERATION
    ARGUMENT            RemoveEntryArgument
    RESULT   RemoveEntryResult
    ERRORS {
        NameError,
        ServiceError, Referral, SecurityError,
        UpdateError}
```

```
RemoveEntryArgument    ::=    OPTIONALLY-SIGNED SET {
        object    [0]         DistinguishedName,
        COMPONENTS OF CommonArguments }

RemoveEntryResult      ::=    NULL
```

11.2.2 Los diversos argumentos tienen los significados definidos en los § 11.2.2.1 a 11.2.2.2.

11.2.2.1 El argumento **object** identifica el asiento a borrar. Los alias en el nombre no son desreferenciados.

11.2.2.2 Los **CommonArguments** (véase el § 7.3) incluyen una especificación de los controles de servicio aplicables a la petición. A los fines de esta operación, la opción **dontDereferenceAlias** y el componente **sizeLimit** son intranscendentes y se ignoran si son proporcionados. Los alias nunca son desreferenciados por esta operación.

11.2.3 Si la petición tiene éxito, se retorna un resultado, aunque no se transportará ninguna información junto con él.

11.2.4 Si la petición fracasa, se informa uno de los errores enumerados. Las circunstancias en las cuales se informarán los distintos errores se definen en el § 12.

11.3 *Modificar asiento (Modify Entry)*

11.3.1 La operación de **ModifyEntry** se usa para realizar una serie de una o más de las siguientes modificaciones a un solo asiento:

a) añadir un nuevo atributo;

b) suprimir un atributo;

c) añadir valores de atributo;

d) suprimir valores de atributo;

e) sustituir valores de atributo;

f) modificar alias.

Los argumentos de la operación pueden ser firmados opcionalmente (véase el § 7.10) por el solicitante.

```
ModifyEntry    ::=    ABSTRACT-OPERATION
    ARGUMENT          ModifyEntryArgument
    RESULT            ModifyEntryResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, SecurityError,
        UpdateError }

ModifyEntryArgument    ::=        OPTIONALLY-SIGNED SET {
    object          [0]    DistinguishedName,
    changes         [1]    SEQUENCE OF EntryModification,
    COMPONENTS OF CommonArguments }

ModifyEntryResult  ::=    NULL

EntryModification  ::=    CHOICE {
    addAttribute    [0]         Attribute,
    removeAttribute             [1]    AttributeType,
    addValues                   [2]    Attribute,
    removeValues    [3]         Attribute }
```

11.3.2 Los diversos argumentos tienen los significados definidos en los § 11.3.2.1 y 11.3.2.2.

11.3.2.1 El argumento **object** identifica el asiento al cual deben aplicarse las modificaciones. Los eventuales alias en el nombre no serán desreferenciados.

11.3.2.2 El argumento **changes** define una secuencia de modificaciones, que se aplican en el orden especificado. Si cualquiera de las modificaciones individuales falla, se genera un **AttributeError** y el asiento queda en el mismo estado en que estaba antes de la operación. Esto es, la operación es atómica. El resultado final de la secuencia de modificaciones no puede violar el esquema de la guía. Sin embargo, es posible, y necesario a veces, que aparezcan cambios individuales de **EntryModification**. Los siguientes tipos de modificación pueden ocurrir:

a) **addAttribute**: Identifica un nuevo atributo que se añadirá al asiento, el cual está completamente especificado por el argumento. Todo intento de añadir un atributo ya existente produce un **AttributeError**.

b) **removeAttribute**: El argumento identifica (por su tipo) un atributo que se quiere eliminar del asiento. Todo intento de eliminar un atributo no existente produce un **AttributeError**.

*Nota* - Esto no se permite si el tipo de atributo está presente en el NDR.

c) **addValues**: Identifica un atributo por el tipo de atributo en el argumento, y especifica uno o más valores de atributos a añadir al atributo. Todo intento de añadir un valor ya existente produce un error. Una tentativa de añadir un valor a un tipo inexistente produce un error.

d) **removeValues**: Identifica un atributo por el tipo de atributo en el argumento, y especifica uno o más valores de atributo a eliminar del atributo. Si los valores no están presentes en el atributo, se produce un **AttributeError**. Si se trata de modificar el atributo de clase de objeto, se retoma un error de actualización.

*Nota* - Esta operación no se permite si uno de los valores está presente en el NDR.

Los valores pueden ser sustituidos por una combinación de **addValues** y **removeValues** en una sola operación **ModifyEntry**.

11.3.2.3 Los **CommonArguments** (véase el § 7.3) incluyen una especificación de los controles de servicio aplicables a la petición. A los fines de esta operación, la opción **dontDereferenceAlias** y el componente **sizeLimit** son intranscendentes y serán ignorados si se suministran. Los alias nunca serán desreferenciados por esta operación.

11.3.3 Si la petición tiene éxito, se retornará un resultado aunque no se transporte información con el mismo.

11.3.4 Si la solicitud fracasa, se informará uno de los errores enumerados. Las circunstancias en las cuales se informan los distintos errores se definen en el § 12.

## 11.4 *Modificar NDR (Modify RND)*

11.4.1 La operación **ModifyRND** se utiliza para cambiar el nombre distinguido relativo de un asiento constitutivo de hoja (un asiento de objeto o un asiento de alias) en el AIG. Opcionalmente, los argumentos de la operación pueden ser firmados (véase el § 7.10) por el solicitante.

```
ModifyRDN ::=          ABSTRACT-OPERATION
    ARGUMENT           ModifyRDNArgument
    RESULT             ModifyRDNResult
    ERRORS {
        NameError,
        ServiceError, Referral, SecurityError,
        UpdateError }

ModifyRDNArgument ::= OPTIONALLY-SIGNED SET {
    object          [0]  DistinguishedName,
    newRDN          [1]  RelativeDistinguishedName,
    deleteOldRDN    [2]  BOOLEAN DEFAULT FALSE,
    COMPONENTS OF CommonArguments }

ModifyRDNResult    ::=  NULL
```

11.4.2 Los diversos parámetros tienen los significados definidos en los § 11.4.2.1 a 11.4.2.5.

11.4.2.1 El argumento **object** identifica el asiento cuyo nombre distinguido relativo se va a modificar. Los alias en el nombre no serán desreferenciados. El asiento superior inmediato no tendrá referencias subordinadas no específicas (véase la Recomendación X.518).

11.4.2.2 El argumento **newRDN** especifica el nuevo NDR del asiento.

11.4.2.3 Si un valor de atributo en el nuevo NDR no existe ya en el asiento (sea como parte del NDR antiguo o como un valor no distinguido), será añadido. Si no puede añadirse, se retorna un error.

11.4.2.4 Si la bandera **deleteOldRDN** está fijada, todos los valores de atributo en el antiguo NDR que no están en el nuevo NDR serán borrados. Si esta bandera no está puesta, los valores antiguos deberán permanecer en el asiento (no como parte del NDR). Se fijará la bandera cuando la operación cambie el valor único de un atributo, en el NDR. Si esta operación suprime el valor perdido de atributo de un atributo, se borrará dicho atributo.

11.4.2.5 Los **CommonArguments** (véase el § 7.3) incluyen una especificación de los controles de servicios aplicables a la petición. A los fines de esta operación, la opción, **dontDereferenceAlias** y el componente **sizeLimit** son intranscendentes y serán ignorados si se suministran. Los alias nunca son desreferenciados por esta operación.

11.4.3 Si la petición tiene éxito, se retorna un resultado, aunque no se transporta información con el mismo.

11.4.4 Si la petición fracasa, se informa uno de los errores enumerados. Las circunstancias en las que deberán devolverse los distintos errores se definen en el § 12.

11.4.5 Como se define en esta Recomendación, esta operación sólo puede utilizarse sobre un asiento constitutivo de hoja.

## 12 Errores

### 12.1 *Precedencia de error*

12.1.1 La guía no continúa realizando una operación más allá del punto en que ella determina que se debe informar un error.

*Nota 1* - Una implicación de esta regla es que el primer error encontrado puede diferir para instancias repetidas de la misma indagación, y no hay un orden lógico específico en el cual procesar una indagación dada. Por ejemplo, los ASG pueden ser buscados en diferentes órdenes.

*Nota 2* - Las reglas de precedencia de error aquí especificadas se aplican solamente al servicio abstracto suministrado por la guía como un todo. Se aplican reglas diferentes cuando se tiene en cuenta la estructura interna de la guía.

12.1.2 Si la guía detecta más de un error simultáneamente, la siguiente lista determina cuál error se informa. Un error en una posición más alta de la lista tiene una precedencia lógica mayor que uno que esté en una posición inferior y es el error que se informa.

a) **NameError**

b) **UpdateError**

c) **AttributeError**

d) **SecurityError**

e) **ServiceError**

12.1.3 Los errores siguientes no presentan conflictos de precedencia:

a) **AbandonFailed**, porque es específico a una operación, **abandon**, la cual no puede sufrir otro error que no sea ese;

b) **Abandoned**, el cual no se informa si una operación de **abandon** se recibe simultáneamente con la detección de un error. En este caso un error de **AbandonFailed**, que informa el problema **tooLate** se informa junto con el informe del error actual encontrado;

c) **Referral**, el cual no es un error "real", sino sólo una indicación de que la guía ha detectado que el AUG tiene que presentar su solicitud a otro punto de acceso.

### 12.2 *Abandonado (Abandoned)*

12.2.1 Este resultado puede ser informado para cualquier operación pendiente de encuesta en la guía (esto es, **Read**, **Search**, **Compare**, **List**) si el AUG invoca una operación **Abandon** con el **InvokeID** apropiado.

**Abandoned** ::= ABSTRACT-ERROR -- *no literalmente un "error"*

12.2.2 Este error no tiene parámetros asociados.

### 12.3 *Fallo de Abandono (Abandon Failed)*

12.3.1 El error **AbandonFailed** informa un problema encontrado durante un intento de abandonar una operación.

```
AbandonFailed      ::=  ABSTRACT-ERROR
    PARAMETER SET {
        problem        [0]  AbandonProblem,
        operation      [1]  InvokeID}

AbandonProblem     ::=  INTEGER
                        noSuchOperation (1),
                        tooLate (2),
                        cannotAbandon (3) }
```

12.3.2 Los diversos parámetros tienen los significados definidos en los § 12.3.2.1 y 12.3.2.2.

12.3.2.1 El **problema** particular encontrado se especifica. Se puede indicar uno de los siguientes problemas:

a) **noSuchOperation**, cuando la guía no tiene conocimiento de la operación que se desea abandonar (esto puede deberse a que no se invocó la operación o a que la guía la ha olvidado);

b) **tooLate**, cuando la guía ya ha respondido a la operación;

c) **cannotAbandon**, cuando se ha tratado de abandonar una operación para lo cual esto está prohibido (por ejemplo, modificar), o no se puede efectuar el abandono.

12.3.2.2 La identificación de la **operación** (invocación) a abandonar.

12.4 *Error de atributo (AttributeError)*

12.4.1 Un **AttributeError** informa un problema relacionado con un atributo.

```
AttributeError  ::=  ABSTRACT-ERROR
    PARAMETER SET {
        object     [0] Name,
        problems   [1] SET OF SEQUENCE {
            problem    [0]  AttributeProblem,
            type       [1]  AttributeType,
            value      [2]  AttributeValue
                                OPTIONAL }}

AttributeProblem    ::=  INTEGER {
    noSuchAttributeOrValue (1),
    InvalidAttributeSyntax (2),
    undefinedAttributeType (3),
    InappropriateMatching (4),
    constraintViolation (5)
    attributeOrValueAlreadyExists (6) }
```

12.4.2 Los diversos parámetros tienen los significados descritos en los § 12.4.2.1 y 12.4.2.2.

12.4.2.1 El parámetro **object** identifica el asiento al que se aplicaba la operación cuando ocurrió el error.

12.4.2.2 Se puede especificar uno o más **problemas**. Cada **problema** identificado más abajo va acompañado de una indicación del **tipo** de atributo y, si es necesario para salvar la ambigüedad, el **valor**, que causó el problema.

a) **noSuchAttributeOrValue**: Al asiento denominado le falta uno de los atributos o valores de atributo especificados como un argumento de la operación;

b) **invalidAttributeSyntax**: Un valor de atributo contemplado, especificado como un argumento de la operación, no es conforme a la sintaxis de atributo del tipo de atributo;

c) **undefinedAttributeType**: Un tipo de atributo no definido fue proporcionado como argumento de la operación. Este error puede ocurrir solamente en relación con las operaciones **Add**, **Remove**, **Modify** o **ModifyRDN**.

d) **inappropiateMatching**: Este caso se da cuando, por ejemplo, en un filtro se trata de utilizar una regla de concordancia no definida para el tipo de atributo en cuestión;

e) **constraintViolation**: Un atributo o valor de atributo suministrado en el argumento de una operación abstracta cumple las restricciones impuestas por la Recomendación X.501 o por la definición del atributo (por ejemplo, el valor es superior al tamaño máximo autorizado);

f) **attributeOrValueAlreadyExists**: Se intentó añadir un atributo que ya existía en el asiento, o un valor que ya existía en el atributo.

12.5 *Error de Nombre (Name Error)*

12.5.1 Un **NameError** informa un problema relacionado al nombre suministrado como un argumento a una operación.

```
NameError      ::=  ABSTRACT-ERROR
     PARAMETER SET {
          problem   [0]  NameProblem,
          matched   [1]  Name}

NameProblem    ::=  INTEGER {
     noSuchObject (1),
     aliasProblem (2),
     invalidAttributeSyntax (3),
     aliasDereferencingProblem (4) }
```

12.5.2 Los diversos parámetros tienen los significados descritos en los § 12.5.2.1 y 12.5.2.2.

12.5.2.1 El **problem** particular encontrado. Cualquiera de los siguientes problemas puede ser indicado:

a) **noSuchObject**: El nombre suministrado (o el nombre resultante, después de la desreferenciación del alias) no concuerda con el nombre de ningún objeto;

b) **aliasProblem**: Se ha desreferenciado un alias que no denomina ningún objeto;

c) **invalidAttributeSyntax**: Un tipo de atributo y su valor de atributo acompañante en un AVA en el nombre son incompatibles;

d) **aliasDereferencingProblem**: Se encontró un alias en una situación en que no estaba autorizado.

12.5.2.2 El parámetro **concordado** contiene el nombre del asiento más bajo (objeto o alias) en el AIG que fue concordado, y es una forma truncada del nombre suministrado, o si un alias ha sido desreferenciado, del nombre resultante.

*Nota* - Si hay un problema relativo a los tipos y/o valores de atributo en el nombre ofrecido en un argumento de una operación de la guía, dicho problema se informa mediante **NameError** (indicándose como problema **invalidAttributeSyntax**) y no un **AttributeError** o un **updateError**.

12.6 *Referimiento (Referral)*

12.6.1 Un **Referral** redirige el usuario del servicio a uno o más puntos de acceso mejor equipados para llevar a cabo la operación solicitada.

```
Referral  ::=  ABSTRACT-ERROR  -- no literalmente un "error"
     PARAMETER SET {
          candidate      [0]  ContinuationReference }
```

12.6.2 El error tiene sólo un parámetro que contiene **ContinuationReference** que pueden utilizarse para hacer progresar la operación (véase la Recomendación X.518).

12.7 *Error de Seguridad (Security Error)*

12.7.1 Un **SecurityError** informa sobre un problema en la ejecución de una operación por razones de seguridad.

```
SecurityError  ::=  ABSTRACT-ERROR
     PARAMETER SET {
          problem   [0]  SecurityProblem }

SecurityProblem     ::=  INTEGER {
     InappropriateAuthentication (1),
     InvalidCredentials (2),
     InsufficientAccessRights (3),
     InvalidSignature (4),
     protectionRequired (5),
     noInformation (6) }
```

12.7.2 El error tiene un solo parámetro que informa sobre el **problema** encontrado. Pueden indicarse los siguientes problemas:

a) **inappropriateAuthentication**: El nivel de seguridad asociado o las credenciales del solicitante no es coherente con el nivel de protección solicitado, por ejemplo, se suministraron credenciales simples cuando se requerían credenciales fuertes;

b) **invalidCredentials**: Las credenciales suministradas no eran válidas;

c) **insufficientAccessRights**: El solicitante no tiene derecho a ejecutar la operación solicitada;

d) **invalidSignature**: Se determinó que la firma del solicitante no era válida;

e) **protectionRequired**: La guía no deseó realizar la operación porque el argumento no estaba firmado.

f) **noInformation**: La operación requerida produjo un error de seguridad para el que no se dispone de información.

12.8 *Error de Servicio (Service Error)*

12.8.1 Un **ServiceError** informa sobre un problema relacionado con la prestación del servicio.

```
ServiceError    ::=    ABSTRACT-ERROR
    PARAMETER SET {
        problem    [0]    ServiceProblem }

ServiceProblem ::=    INTEGER {
            busy (1),
            unavailable (2),
            unwillingToPerform (3),
            chainingRequired (4),
            unableToProceed (5),
            invalidReference (6),
            timeLimitExceeded (7),
            administrativeLimitExceeded (8),
            loopDetected (9),
            unavailableCriticalExtension (10),
            outOfScope (11),
            ditError (12) }
```

12.8.2 El error tiene un solo parámetro, que informa sobre el **problema** particular encontrado. Pueden indicarse los siguientes problemas:

a) **busy**: La guía, o alguna parte de la misma, está en ese momento demasiado ocupada para realizar la operación solicitada pero podrá hacerlo en breve plazo;

b) **unavailable**: La guía, o alguna parte de la misma, no está disponible en ese momento;

c) **unwillingToPerform**: La guía o una parte de la misma no está preparada para ejecutar esta petición, por ejemplo, si se necesita una gran cantidad de recursos para atender a esa petición o si ésta entraña una violación de la política de la autoridad administrativa en cuestión;

d) **chainingRequired**: La guía no tiene otra manera de satisfacer la respuesta que mediante una concatenación, la cual está prohibida por medio de la opción de control del servicio **concatenaciónProhibida**;

e) **unableToProceed**: El ASG que devuelve este error no tenía autoridad administrativa para el contexto de denominación apropiado y, en consecuencia, estaba incapacitado para participar en la resolución de un nombre;

f) **invalidReference**: El ASG fue incapaz de realizar la solicitud dirigida por el AUG (a través del **OperationProgress**). Esto puede producirse debido a un reenvío no válido.

g) **timeLimitExceeded**: La guía ha alcanzado el límite de tiempo establecido por el usuario en un control de servicio. No se retornan resultados parciales al usuario;

h) **administrativeLimitExceeded**: La guía ha llegado al límite establecido por una autoridad administrativa; no se comunican resultados parciales al usuario;

i) **loopDetected**: El ASG fue incapaz de realizar la solicitud dirigida por el AUG (a través del **OperationProgress**). Esto puede producirse debido a un reenvío no válido. La guía es incapaz de satisfacer la solicitud debido a un bucle interno;

j) **unavailableCriticalExtension**: La guía no pudo ejecutar la petición porque una o más de las ampliaciones críticas no estaban disponibles;

k) **outOfScope**: No hubo reenvíos disponibles durante el alcance solicitado.

l) **ditError**: La guía es incapaz de satisfacer la solicitud debido a un problema de coherencia del AIG.

12.9 *Error de actualización (Update Error)*

12.9.1 Un **UpdateError** informa sobre problemas relacionados con tentativas de añadir, suprimir o modificar información en la BIG.

```
UpdateError    ::=   ABSTRACT-ERROR
    PARAMETER SET {
        problem   [0]   UpdateProblem }

UpdateProblem ::=   INTEGER {
    namingViolation (1),
    objectClassViolation (2),
    notAllowedOnNonLeaf (3),
    notAllowedOnRDN (4),
    entryAlreadyExists (5),
    affectsMultipleDSAs (6),
    objectClassModificationProhibited (7) }
```

12.9.2 El error tiene un parámetro **problem** único, que informa el **problem** particular con que se tropezó. Pueden indicarse los siguientes problemas:

a) **namingViolation**: La adición o modificación intentada violaría las reglas de estructuración del AIG definidos en el esquema de la guía y en la Recomendación X.501. Esto es, situaría un asiento como el subordinado de un asiento de alias o en una región del AIG en la cual no está permitido ningún miembro de esa clase de objeto, o definiría un DNR para un asiento que incluyera un tipo de atributo prohibido;

b) **objectClassViolation**: La actualización intentada produciría un asiento inconsecuente con la definición proporcionada por su clase de objeto o con una definición de la Recomendación X.501 en lo que ésta es aplicable a las clases de objeto;

c) **notAllowedOnNonLeaf**: La operación intentada sólo está autorizada en asientos que constituyen hojas del AIG;

d) **notAllowedOnRDN**: La operación intentada aceptaría al DNR (por ejemplo, supresión de un atributo que forma parte del DNR);

e) **entryAlreadyExists**: Una operación de añadir asiento intentada nombra un asiento que ya existe;

f) **affectsMultipleDSAs**: Una actualización intentada necesitaría operar sobre múltiples ASG, lo que no se permite;

g) **objectClassModificationProhibited**: Una operación intentó modificar el atributo de clase de objeto.

*Nota* - El **UpdateError** no se utiliza para informar sobre problemas relacionados con los tipos de atributo, los valores, o las violaciones de limitaciones encontradas con las operaciones **AddEntry**, **RemoveEntry**, **ModifyEntry**, o **ModifyRDN**. Estos problemas se informan mediante un **AttributeError**.

## ANEXO A

(a la Recomendación X.511)

### Servicio abstracto en NSA.1

Este anexo forma parte de la Recomendación.

Este anexo incluye todas las definiciones de tipo, valor y macro NSA.1 contenidas en esta Recomendación en forma del módulo NSA.1 **DirectoryAbstractService**.

```
DirectoryAbstractService {joint-ISO-CCITT ds(5) modules(1) directoryAbstractService(2)}
DEFINITIONS ::=
BEGIN

EXPORTS
     directory, readPort, searchPort, modifyPort,
     DirectoryBind, DirectoryBindArgument,
     DirectoryUnbind,
     Read, ReadArgument, ReadResult,
     Abandon, AbandonArgument, AbandonResult,
     Compare, CompareArgument, CompareResult,
     List, ListArgument, ListResult,
     Search, SearchArgument, SearchResult,
     AddEntry, AddEntryArgument, AddEntryResult,
     RemoveEntry, RemoveEntryArgument, RemoveEntryResult,
     ModifyEntry, ModifyEntryArgument, ModifyEntryResult,
     ModifyRDN, ModifyRDNArgument, ModifyRDNResult,
     Abandoned, AbandonFailed, AttributeError, NameError,
     Referral, SecurityError, ServiceError, UpdateError,
     SecurityParameters;

IMPORTS
     informationFramework, authenticationFramework,
     distributedOperations, directoryObjectIdentifiers
          FROM    UsefulDefinitions    {joint-iso-ccitt ds(5) modules(1)
                                        usefulDefinitions(0)}

OBJECT, PORT, ABSTRACT-BIND, ABSTRACT-UNBIND,
ABSTRACT-OPERATION, ABSTRACT-ERROR
     FROM    AbstractServiceNotation    {joint-iso-ccitt mhs-motis(6)
                                        asdc(2) modules(0) notation(1) }

Attribute, AttributeType, AttributeValue, AttributeValueAssertion,
DistinguishedName, Name, RelativeDistinguishedName
     FROM    InformationFramework InformationFramework

id-ot-directory, id-ot-dua, id-pt-read, id-pt-search, id-pt-modify
     FROM    DirectoryObjectIdentifiers  directoryObjectIdentifiers

ContinuationReference, OperationProgress
     FROM    DistributedOperations distributedOperations

Certificate, CertificationPath, SIGNED,
PROTECTED, AlgorithmIdentifier
     FROM    AuthenticationFramework authenticationFramework
InvokeID,
     FROM    Remote-Operations-Notation    {joint-iso-ccitt
                                          remoteOperations(4) notation(0)};


-- macro para representar firmado opcional --

OPTIONALLY-SIGNED MACRO ::=
BEGIN
     TYPE NOTATION    ::=    type (Type)
     VALUE NOTATION   ::=    value (VALUE CHOICE { Type, SIGNED Type})
END


-- objetos y puertos --

directory
     OBJECT
          PORTS  { readPort [S].
                   searchPort [S],
                   modifyPort [S]}
::=  id-ot-directory
```

```
dua
     OBJECT
          PORTS  { readPort [C],
                   searchPort [C]
                   modifyPort [C]}
::=  id-ot-dua

readPort
     PORT
          CONSUMER INVOKES {
               Read, Compare, Abandon}
::=  id-pt-read

searchPort
     PORT
          CONSUMER INVOKES {
               List, Search }
::=  id-pt-search

modifyPort
     PORT
          CONSUMER INVOKES {
               AddEntry, RemoveEntry,
               ModifyEntry, ModifyRDN}
::=  id-pt-modify


-- vinculación y desvinculación --


DirectoryBind  ::=  ABSTRACT-BIND
     TO { readPort, searchPort, modifyPort }
     BIND
     ARGUMENT   DirectoryBindArgument
     RESULT     DirectoryBindResult
     BIND-ERROR DirectoryBindError

DirectoryBindArgument  ::=       SET {
     credentials      [0]  Credentials  OPTIONAL,
     versions         [1]  Versions DEFAULT v1988}

Credentials  ::=      CHOICE {
     simple     [0]  SimpleCredentials,
     strong     [1]  StrongCredentials,
     externalProcedure  [2]  EXTERNAL }

SimpleCredentials    ::=  SEQUENCE {
     name [0]  DistinguishedName,
     validity  [1]  SET {
          time1     [0]  UTCTime OPTIONAL,
          time2     [1]  UTCTime OPTIONAL,
          random1   [2]  BIT STRING OPTIONAL,
          random2   [3]  BIT STRING OPTIONAL }
                                        OPTIONAL,
          password  [2]  OCTET STRING OPTIONAL }

StrongCredentials    ::=  SET {
     certification-path        [0]  CertificationPath OPTIONAL,
     bind-token                [1]  Token }

Token     ::=  SIGNED SEQUENCE {
     algorithm [0]  AlgorithmIdentifier
     name      [1]  DistinguishedName,
     time      [2]  UTCTime,
     random    [3]  BIT STRING }

Versions ::= BIT STRING (v1988(0))

DirectoryBindResult  ::= DirectoryBindArgument
```

```
DirectoryBindError  ::=  SET {
     versions [0] Versions DEFAULT v1988,
     CHOICE {
          serviceError    [1]  ServiceProblem,
          securityError   [2]  SecurityProblem }}

DirectoryUnbind     ::=  ABSTRACT-UNBIND
     FROM {readPort, searchPort, modifyPort }

-- operaciones, argumentos y resultados --

Read ::=  ABSTRACT-OPERATION
     ARGUMENT  ReadArgument
     RESULT    ReadResult
     ERRORS {
          AttributeError, NameError,
          ServiceError, Referral, Abandoned,
          SecurityError }

ReadArgument ::=  OPTIONALLY-SIGNED SET {
     object          [0]  Name,
     selection       [1]  EntryInformationSelection
                               DEFAULT {},
     COMPONENTS OF CommonArguments }

ReadResult     ::=        OPTIONALLY-SIGNED SET {
     entry           [0]  EntryInformation,
     COMPONENTS OF CommonResults }

Compare ::=  ABSTRACT-OPERATION
     ARGUMENT        CompareArgument
     RESULT          CompareResult
     ERRORS {
          AttributeError, NameError,
          ServiceError, Referral, Abandoned,
          SecurityError }

CompareArgument  ::=  OPTIONALLY-SIGNED SET {
     object          [0]  Name,
     purported       [1]  AttributeValueAssertion,
     COMPONENTS OF CommonArguments }

CompareResult ::=  OPTIONALLY-SIGNED SET {
     DistinguishedName OPTIONAL,
     matched [0]  BOOLEAN,
     fromEntry [1] BOOLEAN DEFAULT TRUE,
     COMPONENTS OF CommonResults }

Abandon ::=  ABSTRACT-OPERATION
     ARGUMENT        AbandonArgument
     RESULT  AbandonResult
     ERRORS  {AbandonFailed}

AbandonArgument ::= SEQUENCE {
     InvokeID [0]  InvokeID}

AbandonResult ::=  NULL

List ::= ABSTRACT-OPERATION
     ARGUMENT        ListArgument
     RESULT     ListResult
     ERRORS {
          AttributeError, NameError,
          ServiceError, Referral, Abandoned,
          SecurityError }

ListArgument  ::=  OPTIONALLY-SIGNED SET {
     object    [0]  Name,
     COMPONENTS OF CommonArguments }
```

```
ListResult      ::=   OPTIONALLY-SIGNED CHOICE{
    listInfo SET    {
    DistinguishedName OPTIONAL
    subordinates [1] SET OF SEQUENCE {
        RelativeDistinguishedName,
        aliasEntry      [0]   BOOLEAN DEFAULT FALSE,
        fromEntry       [1]   BOOLEAN DEFAULT TRUE },
        partialOutcomeQualifier [2] PartialOutcomeQualifier
                                                    OPTIONAL,
        COMPONENTS OF CommonResults},
    uncorrelatedListInfo [0] SET OF
        ListResult }

PartialOutcomeQualifier   ::=   SET {
    limitProblem    [0]   LimitProblem OPTIONAL,
    unexplored      [1]   SET OF
        ContinuationReference OPTIONAL,
    unavailableCriticalExtensions [2] BOOLEAN DEFAULT FALSE }

LimitProblem   ::=   INTEGER {
    timeLimitExceeded   (0),
    sizeLimitExceeded   (1),
    administrativeLimitExceeded (2) }

Search    ::=   ABSTRACT-OPERATION
    ARGUMENT            SearchArgument
    RESULT              SearchResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, Abandoned,
        SecurityError }

SearchArgument ::=  OPTIONALLY-SIGNED SET {
    baseObject      [0]   Name,
    subset          [1]   INTEGER {
        baseObject(0),
        oneLevel(1),
        wholeSubtree(2)} DEFAULT baseObject,
    filter          [2]   Filter DEFAULT and {},
    searchAliases   [3]   BOOLEAN DEFAULT TRUE,
    selection       [4]   EntryInformationSelection DEFAULT {},
    COMPONENTS OF CommonArguments }

SearchResult    ::=   OPTIONALLY-SIGNED
    CHOICE {
    searchInfo SET {
    DistinguishedName OPTIONAL,
    entries  [0]  SET OF EntryInformation,
    partialOutcomeQualifier
        [2] partialOutcomeQualifier OPTIONAL,
    COMPONENTS OF CommonResults },
    uncorrelatedSearchInfo [0] SET OF
        SearchResult }

AddEntry ::=   ABSTRACT-OPERATION
    ARGUMENT        AddEntryArgument
    RESULT      AddEntryResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, SecurityError,
        UpdateError }

AddEntryArgument  ::=   OPTIONALLY-SIGNED SET {
        object      [0]         DistinguishedName,
        entry       [1]         SET OF Attribute,
        COMPONENTS OF CommonArguments}

AddEntryResult ::= NULL
```

```
RemoveEntry ::=          ABSTRACT-OPERATION
    ARGUMENT             RemoveEntryArgument
    RESULT               RemoveEntryResult
    ERRORS {
        NameError,
        ServiceError, Referral, SecurityError,
        UpdateError}

RemoveEntryArgument ::= OPTIONALLY-SIGNED SET {
        object    [0]  DistinguishedName,
        COMPONENTS OF CommonArguments }

RemoveEntryResult ::= NULL

ModifyEntry ::=      ABSTRACT-OPERATION
    ARGUMENT         ModifyEntryArgument
    RESULT           ModifyEntryResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, SecurityError,
        UpdateError}

ModifyEntryArgument ::= OPTIONALLY-SIGNED SET {
    object    [0]  DistinguishedName,
    changes   [1]  SEQUENCE OF EntryModification,
    COMPONENTS OF CommonArguments }

ModifyEntryResult ::= NULL

EntryModification ::= CHOICE {
    addAttribute        [0]  Attribute,
    removeAttribute     [1]  AttributeType,
    addValues           [2]  Attribute,
    removeValues        [3]  Attribute}

ModifyRDN ::=        ABSTRACT-OPERATION
    ARGUMENT         ModifyRDNArgument
    RESULT           ModifyRDNResult
    ERRORS {
        NameError,
        ServiceError, Referral, SecurityError,
        UpdateError }

ModifyRDNArgument ::= OPTIONALLY-SIGNED SET {
    object          [0]      DistinguishedName,
    newRDN          [1]      RelativeDistinguishedName,
    deleteOldRDN    [2]      BOOLEAN DEFAULT FALSE,
    COMPONENTS OF CommonArguments }

ModifyRDNResult ::=          NULL


-- errores y parámetros --


Abandoned ::= ABSTRACT-ERROR        -- no literalmente un "error"

AbandonFailed ::= ABSTRACT-ERROR
    PARAMETER SET {
        problem         [0]  AbandonProblem,
        operation       [1]  InvokeID}

AbandonProblem ::= INTEGER {
                   noSuchOperation (1),
                   tooLate (2),
                   cannotAbandon (3)}
```

```
AttributeError ::= ABSTRACT-ERROR
    PARAMETER SET {
        object          [0]  Name,
        problems        [1]  SET OF SEQUENCE {
            problem          [0]  AttributeProblem,
            type             [1]  AttributeType,
            value            [2]  AttributeValue OPTIONAL }}

AttributeProblem ::=
    INTEGER {
        noSuchAttributeOrValue (1),
        invalidAttributeSyntax (2),
        undefinedAttributeType (3),
        inappropriateMatching (4),
        constraintViolation (5),
        attributeOrValueAlreadyExists (6) }

NameError ::= ABSTRACT-ERROR
    PARAMETER SET {
        problem  [0]  NameProblem,
        matched  [1]  Name}

NameProblem ::= INTEGER {
        noSuchObject (1),
        aliasProblem (2),
        invalidAttributeSyntax (3),
        aliasDereferencingProblem (4)}

Referral ::= ABSTRACT-ERROR    -- no literalmente un "error"
    PARAMETER SET {
        candidate       [0]  ContinuationReference}

SecurityError ::= ABSTRACT-ERROR
    PARAMETER SET {
        problem  [0]  SecurityProblem }

SecurityProblem ::= INTEGER {
    inappropriateAuthentication (1),
    invalidCredentials (2),
    insufficientAccessRights (3),
    invalidSignature (4),
    protectionRequired (5),
    noInformation (6) }

ServiceError ::= ABSTRACT-ERROR
    PARAMETER SET {
        problem  [0]  ServiceProblem }

ServiceProblem ::= INTEGER {
    busy (1),
    unavailable (2),
    unwillingToPerform (3),
    chainingRequired (4),
    unableToProceed (5),
    invalidReference (6),
    timeLimitExceeded (7),
    administrativeLimitExceeded (8),
    loopDetected (9),
    unavailableCriticalExtension (10),
    outOfScope (11),
    ditError (12) }

UpdateError ::= ABSTRACT-ERROR
    PARAMETER SET {
        problem  [0]  UpdateProblem }
```

```
UpdateProblem ::=   INTEGER {
      namingViolation (1),
      objectClassViolation (2),
      notAllowedOnNonLeaf (3),
      notAllowedOnRDN (4),
      entryAlreadyExists (5),
      affectsMultipleDSAs (6),
      objectClassModificationProhibited (7)}

-- argumentos resultados/comunes --

CommonArguments  ::=   SET {
      [30]  ServiceControls DEFAULT {}
      [29]  SecurityParameters DEFAULT {},
      requestor [28] DistinguishedName OPTIONAL,
      [27]  OperationProgress DEFAULT notStarted,
      aliasedRDNs [26]  INTEGER OPTIONAL,
      extensions [25] SET OF Extension OPTIONAL }

Extension       ::=   SET {
      identifier        [0]   INTEGER,
      critical          [1]   BOOLEAN DEFAULT FALSE,
      item              [2]   ANY DEFINED BY identifier }

CommonResults ::=   SET {
      [30]  SecurityParameters  OPTIONAL,
      performer [29] DistinguishedName OPTIONAL,
      aliasDereferenced  [28] BOOLEAN DEFAULT FALSE}

-- tipos de datos comunes --

ServiceControls  ::=          SET {
      options [0]       BIT STRING {
      preferChaining (0),
      chainingProhibited (1),
      localScope (2),
      dontUseCopy (3),
      dontDereferenceAliases(4)}
            DEFAULT{},

      priority [1] INTEGER {
            low (0),
            medium (1),
            high (2) } DEFAULT medium,

timeLimit [2]   INTEGER OPTIONAL,

sizeLimit  [3]   INTEGER OPTIONAL,

scopeOfReferral [4]   INTEGER {
                  dmd(0),
                  country(1)}
                  OPTIONAL }

EntryInformationSelection        ::= SET {
      attributeTypes
            CHOICE {
                  allAttributes [0]  NULL,
                  select [1] SET OF AttributeType
                  -- conjunto vacío implica que no se
                  -- solicitan atributos --}
            DEFAULT allAttributes NULL,

      infoTypes [2] INTEGER {
            attributeTypesOnly (0),
            attributeTypesAndValues (1) } DEFAULT
                        attributeTypesandValues }
```

```
EntryInformation ::=          SEQUENCE {
    DistinguishedName,
    fromEntry BOOLEAN DEFAULT TRUE,
    SET OF CHOICE {
        AttributeType,
        Attribute} OPTIONAL }

Filter      ::=   CHOICE {
    item          [0]   FilterItem,
    and           [1]   SET OF Filter,
    or            [2]   SET OF Filter,
    not           [3]   Filter }

FilterItem        ::=   CHOICE {
    equality          [0]   AttributeValueAssertion,
    substrings        [1]   SEQUENCE {
        type              AttributeType,
        strings           SEQUENCE OF CHOICE {
                          initial     [0]   AttributeValue,
                          any         [1]   AttributeValue,
                          final       [2]   AttributeValue}},
    greaterOrEqual        [2]   AttributeValueAssertion,
        lessOrEqual       [3]   AttributeValueAssertion,
        present           [4]   AttributeType,
    approximateMatch      [5]   AttributeValueAssertion }

SecurityParameters ::=    SET {
    certification-Path    [0]   CertificationPath OPTIONAL,
    name              [1]   DistinguishedName OPTIONAL,
    time              [2]   UTCTime OPTIONAL,
    random            [3]   BIT STRING OPTIONAL,
    target            [4]   ProtectionRequest OPTIONAL }

ProtectionRequest ::=     INTEGER {
                              none(0),
                              signed (1)}
```

## ANEXO B

(a la Recomendación X.511)

### Identificadores de objeto de la guía

Este anexo forma parte de la Recomendación.

Este anexo incluye todos los identificadores de objeto ASN.1 contenidos en la presente Recomendación en forma de módulo ASN.1 "**DirectoryObjectIdentifiers**".

```
DirectoryObjectIdentifiers      {joint-ISO-CCITT ds(5) modules(1)
                                directoryObjectIdentifiers(9)}
DEFINITIONS ::=
BEGIN

EXPORTS
    id-ot-directory, id-ot-dua, id-pt-read, id-pt-search, id-pt-modify;

IMPORTS
    id-ot, id-pt
    FROM UsefulDefinitions  {joint-iso-ccitt ds(5) modules(1),
                            usefulDefinitions(0)};
```

```
-- Objetos --

id-ot-directory      OBJECT IDENTIFIER ::=  {id-ot 1}

id-ot-dua            OBJECT IDENTIFIER ::=  {id-ot 2}

-- Tipos de puerto --

id-pt-read           OBJECT IDENTIFIER ::=  {id-pt 1}

id-pt-search         OBJECT IDENTIFIER ::=  {id-pt 2}

id-pt-modify         OBJECT IDENTIFIER ::=  {id-pt 3}

END
```

**Recomendación X.518**

## LA GUIA - PROCEDIMIENTOS PARA OPERACION DISTRIBUIDA [1)]

*(Melbourne, 1988)*

INDICE



---

1) La Recomendación X.518 y la norma ISO 9594-4, Information Processing Systems - Open Systems Interconnection - The Directory - Procedures for Distributed Operation (Sistemas de procesamiento de información - Interconexión de sistemas abiertos - La guía - Procedimientos para operación distribuida) fueron redactadas en estrecha colaboración y están técnicamente alineadas.

SECCION 1 - *Introducción*

## 0 Introducción

0.1 Este documento, junto con otros de la misma serie, se ha elaborado para facilitar la interconexión de sistemas de procesamiento de información con miras a la prestación de servicios de guía. El conjunto de tales sistemas y la información en ellos contenida puede contemplarse como un todo integrado al que se denomina la guía. La información que figura en la guía, conocida globalmente como base de información de la guía (BIG), tiene como utilización típica la de facilitar las comunicaciones entre, con o a propósito de, objetos tales como entidades de aplicación de ISA, personas, terminales y listas de distribución.

0.2 La guía desempeña un papel significativo en la interconexión de sistemas abiertos cuyo objetivo es permitir, con un mínimo de consenso técnico fuera de las propias normas de interconexión, la interconexión de sistemas de procesamiento de información:

- de diferentes fabricantes;
- sujetos a gestiones diferentes;
- de distintos grados de complejidad; y
- de distintas fechas de construcción.

0.3 En esta Recomendación se especifican los procedimientos según los cuales se establece el interfuncionamiento de los componentes distribuidos de la guía para proporcionar un servicio coherente a sus usuarios.

## 1 Alcance y campo de aplicación

1.1 La presente Recomendación especifica el comportamiento de los ASG que participan en la aplicación guía distribuida. El comportamiento permitido está concebido de manera tal que se asegure un servicio coherente dada una amplia distribución de la BIG a través de muchos ASG.

1.2 No se pretende que la guía sea un sistema de base de datos de uso general, aunque puede fundamentarse en sistemas de ese tipo. Se supone que la frecuencia de las consultas es notablemente superior a la de las actualizaciones.

## 2 Referencias

Recomendación X.200 - Modelo de referencia de interconexión de sistemas abiertos para aplicaciones del CCITT.

Recomendación X.208 - Especificación de notación de sintaxis abstracta uno (NSA.1).

Recomendación X.500 - La guía - Visión de conjunto de conceptos, modelos y servicios.

Recomendación X.501 - La guía - Modelos

Recomendación X.511 - La guía - Definición del servicio abstracto

Recomendación X.519 - La guía - Especificaciones de protocolos

Recomendación X.520 - La guía - Tipos de atributo seleccionados

Recomendación X.521 - La guía - Clases de objeto seleccionados

Recomendación X.407 - Sistema de tratamiento de mensajes: Convenios para la definición del servicio abstracto

## 3 Definiciones

Las definiciones que figuran en este punto utilizan las siglas definidas en el § 4.

3.1 *Definiciones del modelo de referencia de ISA*

Esta Recomendación utiliza el término siguiente, definido en la Recomendación X.200:

a) *título de entidad de aplicación*

3.2 *Definiciones básicas relativas a la guía*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.500:

a) *(la) guía;*

b) *base de información de la guía.*

3.3 *Definiciones relativas al modelo de guía*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.501:

a) *punto de acceso;*

b) *alias;*

c) *nombre distinguido;*

d) *árbol de información de la guía;*

e) *agente de sistema de guía;*

f) *agente de usuario de la guía;*

g) *nombre distinguido relativo.*

3.4 *Convenios de definición de sintaxis abstracta*

Esta Recomendación utiliza los términos siguientes, definidos en la Recomendación X.407.

a) *error abstracto;*

b) *operación abstracta;*

c) *resultado.*

3.5 *Definiciones relativas a la operación distribuida*

En esta Recomendación se utilizan los términos que a continuación se indican y definen:

a) *concatenación:* modo de interacción empleado facultativamente por un ASG que no pueda efectuar una operación por sí mismo. El ASG concatena invocando una operación de otro ASG y transfiriendo después el resultado al solicitante original;

b) *prefijo de contexto:* la secuencia de NDR que va desde la raíz del AIG al vértice inicial de un contexto de denominación; corresponde al nombre distinguido de ese vértice;

c) *referencia recíproca:* una referencia de conocimiento en la que figura información sobre el ASG que contiene un asiento. Se emplea para optimización. El asiento no debe tener relación superior o subordinada;

d) *fragmento de BIG:* la porción de la BIG contenida en un ASG y que abarca uno o más contextos de denominación;

e) *resolución de nombre distribuida:* proceso por el cual la resolución de nombre se efectúa en más de un ASG;

f) *referencia interna:* una referencia de conocimiento en la que hay un puntero interno señalando a un asiento contenido en el mismo ASG;

g) *información de conocimiento:* la información que un determinado ASG posee sobre los asientos que contiene y sobre cómo localizar otros asientos en la guía;

h) *referencia de conocimiento:* conocimiento que asocia directa o indirectamente un asiento de AIG con el ASG en el que está ubicado;

i) *árbol de conocimiento:* modelo conceptual de la información de conocimiento contenido en un ASG que le permite llevar a cabo la resolución de nombre distribuida;

j) *difusión:* un modo de interacción al que puede recurrir, de manera facultativa, un ASG incapaz de efectuar una operación por sí mismo. El ASG *difunde* la operación, es decir, invoca la misma operación en varios otros ASG (en serie o en paralelo) y transfiere un resultado apropiado al solicitante original;

k) *resolución de nombre:* proceso de localización de un asiento por el cual se hace concordar secuencialmente cada nombre distinguido relativo (NDR) de un nombre contemplado con un vértice del árbol de información de la guía (AIG);

l) *contexto de denominación:* un sub-árbol parcial del AIG que comienza en un vértice y se extiende hacia abajo hasta vértices constitutivos y/o no constitutivos de hoja. Tales

vértices forman el límite del contexto de denominación. Los vértices no constitutivos de hoja pertenecientes a ese límite indican el comienzo de otros contextos de denominación;

m) *referencia subordinada no específica:* una referencia de conocimiento que posee información sobre el ASG que contiene uno o más asientos subordinados no especificados;

n) *progresión de la operación:* un conjunto de valores que indica hasta qué punto se ha producido la resolución de nombre;

o) *trayecto de referencia:* una secuencia continua de referencias de conocimiento;

p) *referimiento:* resultado que puede retornar un ASG incapaz de efectuar una operación por sí mismo y que identifica uno o más ASG distintos más capaces de efectuarla;

q) *desglose de una petición:* descomposición de una petición en subpeticiones, cada una de las cuales cumple una parte de la operación distribuida;

r) *contexto de raíz:* el contexto de denominación para el vértice cuyo nombre comprende la secuencia vacía de NDR;

s) *referencia subordinada:* referencia de conocimiento en la que hay información sobre el ASG que contiene un asiento subordinado específico;

t) *subpetición:* una petición generada por desglose de una petición;

u) *referencia superior:* referencia de conocimiento en la que hay información sobre el ASG que contiene un asiento superior.

## 4 Abreviaturas

En la presente Recomendación se utilizan las siguientes abreviaturas:

AIG Arbol de información de la guía

ASG Agente de sistema de guía

AUG Agente de usuario de guía

BIG Base de información de la guía

NDR Nombre distinguido relativo

## 5 Notación

La notación empleada en este punto se define donde a continuación se indica:

a) la notación de sintaxis de datos, la notación de codificación y la notación macro se definen en la Recomendación X.208;

b) las notaciones para modelos y servicios abstractos se definen en la Recomendación X.407.

# SECCION 2 - *Visión de conjunto*

## 6 Visión de conjunto

El servicio abstracto de guía permite la interrogación, la extracción y la modificación de información de guía en la BIG. Este servicio se describe desde el punto de vista del objeto de guía abstracto tal como se define en la Recomendación X.511.

Por necesidad, la especificación del objeto de guía abstracto no se refiere en modo alguno a la realización física de la guía: no se ocupa, en concreto, de la especificación de los agentes de sistema de guía (ASG), en los que se almacena y gestiona la BIG, y a través de los que se presta el servicio. Más aún, esa especificación tampoco tiene en cuenta si la BIG es centralizada, es decir, contenida en un único ASG o distribuida entre un cierto número de ASG. Por consiguiente, las exigencias para que los ASG tengan conocimiento de, naveguen hacia, y cooperen con otros ASG a fin de facilitar el servicio abstracto en un entorno distribuido quedan también fuera del alcance de la descripción del servicio.

En esta Recomendación se especifica el refinamiento del objeto de guía abstracto, viniendo expresado ese refinamiento en base a uno o más objetos de AGS que en conjunto constituyen el servicio de guía distribuida. Estrechamente ligada a esto se halla la identificación y especificación de los puertos de ASG que son internos al objeto de guía. Para cada uno de esos puertos se especifican en esta Recomendación el servicio abstracto asociado y sus procedimientos.

Se especifican además las maneras permitidas de distribuir la BIG entre uno o más ASG. En el caso límite en que la BIG estuviera contenida en un solo ASG, la guía sería de hecho una guía centralizada; para el caso en que la BIG esté distribuida entre dos o más ASG, se especifican mecanismos de conocimiento y navegación que garantizan que la totalidad de la BIG es potencialmente accesible desde todos los ASG que contengan asientos constituyentes.

Asimismo, se especifican interacciones para el tratamiento de solicitudes que permiten el que determinadas características operativas de la guía sean controladas por sus usuarios. En concreto, el usuario controla si un ASG, en respuesta a una indagación de la guía relativa a información contenida en otro u otros ASG, puede interrogar al otro a los otros ASG directamente (concatenación/difusión), o si debe responder con información sobre otro u otros ASG que haga avanzar la investigación (referimiento).

Por lo general, la elección de un ASG entre concatenación/difusión o referimiento, vendrá determinada por los controles de servicio establecidos por el usuario, y por las circunstancias administrativas, operacionales o técnicas propias del ASG.

Habida cuenta de que, por lo general, la guía será distribuida y que las consultas de la guía serán satisfechas por un número arbitrario de ASG cooperantes, que podrán arbitrariamente concatenar/difundir o referir, según los criterios expuestos más arriba, en esta Recomendación se especifican los procedimientos apropiados a que deben atenerse los ASG en respuesta a consultas de una guía distribuida. Dichos procedimientos garantizarán a los usuarios del servicio de guía distribuida un servicio a la vez cómodo y coherente.

## SECCION 3 - *Modelos de guía distribuida*

### 7 Modelo de sistema de guía distribuida

El servicio abstracto de guía, tal como se define en la Recomendación X.511, modela la guía como un objeto que presta un conjunto de servicios de guía a sus usuarios. Los servicios de la guía son modelados en base a puertos, proporcionando cada puerto un determinado conjunto de servicios de guía. Los usuarios de la guía ganan acceso a sus servicios a través de un punto de acceso. La guía puede tener uno o más puntos de acceso, caracterizado cada uno de ellos por los servicios que proporciona y el modo de interacción empleado para proporcionarlos.

En este punto se analiza la estructura interna del objeto de guía, identificándose los objetos que lo componen y sus puertos, con lo que se facilita la especificación de un servicio de guía distribuida.

La figura 1/X.518 ilustra el modelo de guía distribuida que servirá de base para la especificación de los aspectos distribuidos de la guía. El objeto de guía que en ella se muestra consta de un conjunto de uno o más objetos-ASG.

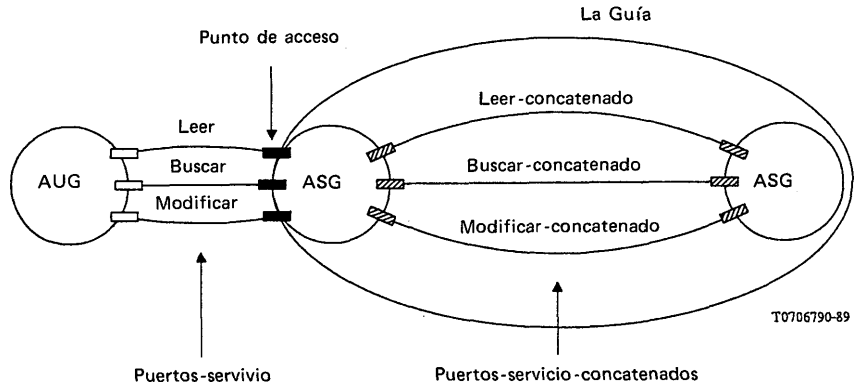



FIGURA 1/X.518

**Objetos del modelo guía distribuida**

En los puntos que siguen se especifican con detalle los objetos de ASG. En éstos se señalan simplemente algunas de sus características, lo que sirve a modo de introducción y para establecer la relación entre la presente y otras Recomendaciones.

Los objetos de ASG se definen para poder acomodar la distribución de la BIG y para que un cierto número de ASG distribuidos físicamente puedan actuar recíprocamente de una manera prescrita y cooperativa proporcionando servicios de guía a los usuarios de la guía (los AUG).

Los objetos de ASG, como los objetos de guía, se caracterizan por sus puertos visibles externamente. Los puertos asociados a un objeto-ASG son de dos tipos: puertos-servicio y puertos-servicio-concatenados.

Los puertos-servicio de un objeto de ASG son idénticos a los del objeto de guía, a saber: puertos de leer **(read)**, buscar **(search)** y modificar **(modify)**. La figura 1/X.518 muestra que los puertos-servicio asociados a un objeto de ASG constituyen un punto-acceso a través del cual se facilitan los servicios de guía.

La especificación detallada de los puertos-servicio **read, search** y **modify** del objeto de ASG se encuentran en la Recomendación X.511. (La especificación de los protocolos de los correspondientes elementos de servicio de aplicación de la ISA, obtenida a partir de estas definiciones de puertos, figura en la Recomendación X.519.)

Además de los puertos-servicio del objeto de ASG que facilitan el acceso al objeto de guía, se define un segundo conjunto de puertos: los puertos-servicio-concatenados. Estos puertos permiten la comunicación entre distintos ASG, de manera que pueda realizarse el servicio abstracto de guía en un entorno distribuido.

Los puertos-servicio-concatenados y las operaciones proporcionadas a través de los mismos se corresponden directamente con los puertos-servicio denominados de manera similar, y son respectivamente, **chainedRead, chainedSearch** y **chainedModify** (respectivamente, Leerconcatenado, Buscarconcatenado y Modificarconcatenado).

El proceso de especificación de los objetos constituyentes de un objeto más abstracto se denomina "refinamiento". La especificación del refinamiento del objeto de guía en sus componentes (los ASG) y la especificación del servicio abstracto proporcionado por cada uno de ellos (el servicio abstracto de ASG) figuran en la Sección Cuatro de la presente Recomendación. (La especificación de los protocolos de los correspondientes elementos de servicio de aplicación de la ISA, tal como se deduce de las definiciones de puertos concatenados, figura en la Recomendación X.519.)

## 8 Modelo de interacciones de los ASG

Una característica básica de la guía es la de que, dada una BIG distribuida, un usuario deba poder obtener la tramitación de cualquier petición de servicio (con las limitaciones inherentes a las políticas de seguridad, de control de acceso y de administración) independientemente de cuál sea el punto de acceso en el que se origine la petición. El cumplimento de este requisito exige que todo ASG que participe en la tramitación de una determinada solicitud de servicio, tenga cierto conocimiento (tal como se especifica en el § 10 de esta Recomendación) de dónde se halla la información pedida y remita ese conocimiento al solicitante o trate en su nombre de que se tramite la petición. (El solicitante puede ser un AUG u otro ASG; en el segundo caso, ambos ASG deben tener un puerto concatenado.)

Se han definido tres modos de interacción de los ASG encaminados a la satisfación de estas exigencias, a saber: "concatenación", "difusión" y "referimiento". La "concatenación" y la "difusión" se definen para satisfacer la segunda, mientras que el "referimiento" trata de cumplir con la primera.

### 8.1 *Concatenación*

Un ASG puede utilizar este modo de interacción (mostrado en la figura 2/X.518) para transferir una petición a otro ASG, cuando el primero tenga conocimiento sobre los contextos de denominación contenidos en el segundo. Es posible emplear la concatenación para contactar con un ASG al que se apunta en una referencia recíproca, subordinada o superior. La difusión, descrita en el § 8.2, es una forma de concatenación.

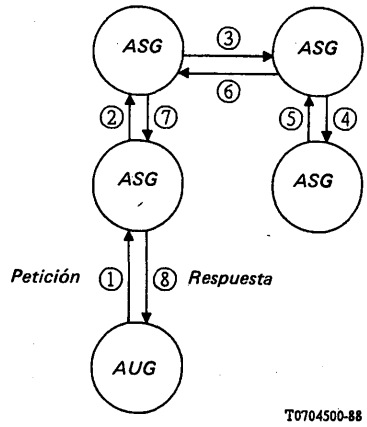



FIGURA 2/X.518

**Modo concatenación**

*Nota* - En la figura 2/X.518, el orden de las interacciones viene dado por los números asociados a las líneas de interacción.

8.2 *Difusión*

Este modo de interacción (mostrado en las figuras 3a/X.512 y 3b/X.518) puede ser utilizado por un ASG para concatenar una petición idéntica en paralelo (a) o en serie (b) a otro u otros ASG, cuando el primero desconoce los contextos de denominación completos contenidos en los segundos. La difusión sólo la utiliza un ASG para contactar otros ASG a los que se apunta en una referencia subordinada no específica. Se pasa la petición idéntica a cada uno de los ASG. Normalmente, durante una resolución de nombre, sólo uno de los ASG podrá continuar el procesamiento de la operación distante; todos los demás retornarán el **error de servicio "no se puede continuar"** (o **"incapaz para proseguir"**). Sin embargo, durante la fase de evaluación de operaciones de búsqueda y listado, todos los ASG en una referencia subordinada no específica deben poder continuar el procesamiento de la petición.

*Nota* - En las figuras 3a/X.518 y 3b/X.518 el orden de las interacciones viene dado por los números asociados a las líneas de interacción.

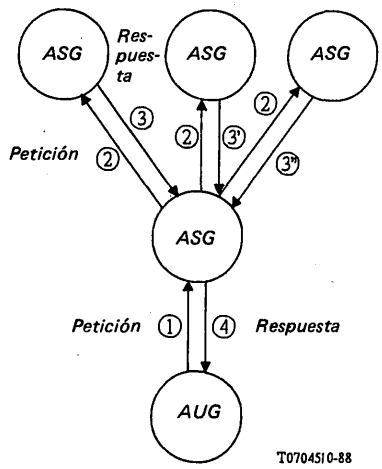



FIGURA 3a/X.518

**Modo difusión**

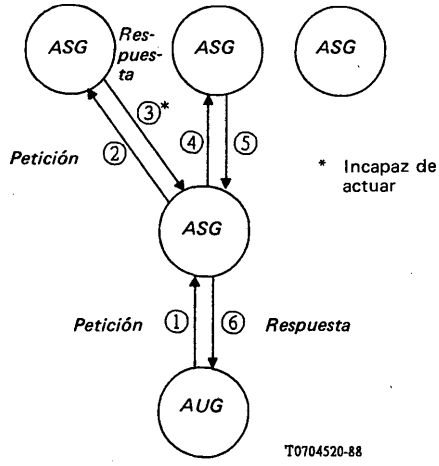



FIGURA 3b/X.518

**Modo difusión**

8.3 *Referimiento*

Un ASG retorna un referimiento (mostrado en las figuras 4a/X.518 y 4b/X.518) en su respuesta a una petición cuya realización le hubiera sido solicitada por un AUG o por otro ASG (en este último caso, ambos ASG deben tener un puerto-servicio-concatenado). Puede ocurrir que por toda respuesta se obtenga simplemente el referimiento (en cuyo caso se le considera un error) o bien que el referimiento sea nada más que una parte de la respuesta. El referimiento contiene una referencia de conocimiento que puede ser superior, subordinada, subordinada recíproca, o subordinada no específica.

El ASG en la figura 4a/X.518 que recibe el referimiento puede utilizar la referencia de conocimiento que él contiene para, a continuación, concatenar o difundir (dependiendo de cuál sea el tipo de referencia) la operación original a otros ASG. Como otra posibilidad, un ASG que recibe un referimiento, puede retransmitirlo, a su vez, en su respuesta. Un AUG (véase la figura 4b/X.518) que reciba un referimiento puede utilizarlo para contactar otro (u otros) ASG para hacer progresar la petición.

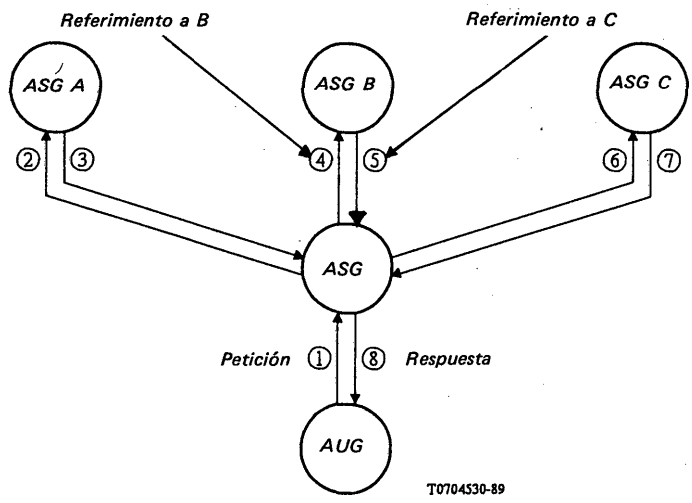



FIGURA 4a/X.518

**Modo referimiento - ASG con puerto concatendado**

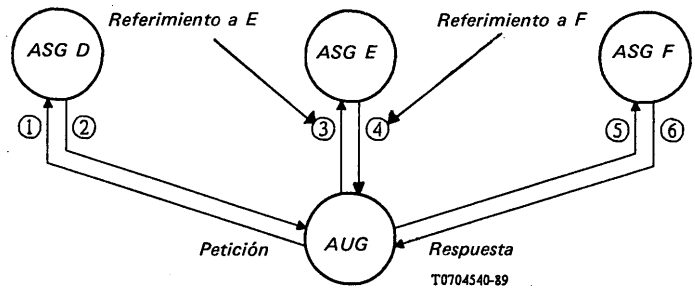



FIGURA 4b/X.518

**Modo referimiento - AUG pide ASG con puertos no concatenados**

*Nota* - En las figuras 4a/X.518 y 4b/X.518, el orden de las interacciones viene dado por los números asociados a las líneas de interacción.

8.4 *Determinación del modo*

Si un ASG no puede por sí solo resolver una petición, debe concatenar/difundir esa petición (o la resultante de su descomposición) a otro ASG, a menos que:

a) la concatenación ha sido prohibida por el usuario mediante los controles de servicio, en cuyo caso el ASG deberá devolver un referimiento o un **error de servicio** de **concatenación requerida** (como último recurso); o

b) el ASG tiene razones administrativas, operacionales o técnicas para preferir no concatenar, en cuyo caso el ASG deberá devolver un referimiento.

*Nota 1* - Una "razón técnica" para no concatenar/difundir es que el ASG identificado en esa referencia de conocimiento no tenga puertos de servicio concatendados.

*Nota 2* - Si el control de servicio **localScope** está fijado, el ASG (o el DGG (dominio de gestión de la guía)) deberán, o bien resolver la petición o retornar un error.

*Nota 3* - Si el usuario prefiere referimientos, deberá fijar **chainingProhibited**.

## 9 Distribución de guía

En este punto se definen los principios según los cuales se puede distribuir la BIG.

Cada asiento de la BIG es administrado por un solo administrador de ASG, al que se atribuye autoridad administrativa sobre ese asiento. El mantenimiento y la gestión deben llevarse a cabo en un ASG administrado por la autoridad administrativa para el asiento.

Aunque la guía no ofrece medios para la replicación de asientos, es posible de todos modos realizar una replicación de dos maneras:

- Almacenando copias de un asiento en otro u otros ASG mediante un acuerdo bilateral. El mantenimiento y la gestión de las copias dependerá de ese acuerdo, y no está definido en la presente Recomendación.
- Almacenando copias de asientos (local y dinámicamente) que se obtienen a partir de copias resultantes de las peticiones.

*Nota* - La adquisición de asientos de cache está sujeto al control de acceso.

El originador de la petición es informado (mediante una **fromcopy**) acerca de si la información retornada en respuesta a una petición se basa o no en un asiento replicado. Se define un control, **dontUseCopy**, que permite al usuario prohibir la utilización de asientos replicados.

Cada ASG de la guía contiene un *fragmento* de la BIG. El fragmento de BIG contenido por un ASG se describe en base al AIG y comprende uno o más contextos de denominación. Un *contexto de denominación* es un sub-árbol parcial del AIG que se identifica comenzando en un vértice y

extendiéndose hacia abajo hasta vértices constitutivos o no constitutivos de hoja. Tales vértices son el límite del contexto de denominación. Los subordinados de los vértices No-hoja pertenecientes a ese límite indican el comienzo de otros contextos de denominación.

Es posible que un administrador de ASG tenga autoridad administrativa para varios contextos de denominación disociados. Por cada contexto de denominación para el que tiene autoridad administrativa, un ASG debe contener, lógicamente, la secuencia de NDR que va desde la raíz del AIG al vértice inicial del sub-árbol que comprenda el contexto de denominación. La secuencia de NDR se llama *prefijo de contexto.*

Un administrador de ASG puede delegar autoridad administrativa a cualesquiera subordinados inmediatos de cualquier asiento contenido localmente respecto, a otro ASG. Un ASG que delega autoridad se denomina *ASG superior* y el contexto que contiene el asiento superior de uno a quien se delegó autoridad administrativa se llama *contexto de denominación superior.* La delegación de la autoridad administrativa comienza en la raíz y desciende en el AIG, es decir, sólo puede tener lugar de un asiento a sus subordinados.

La figura 5/X.518 muestra un AIG hipotético repartido lógicamente entre cinco contextos de denominación (llamados A, B, C, D y E) distribuidos físicamente en tres ASG (ASG1, ASG2 y ASG3).

Según se ve en el ejemplo, los contextos de denominación contenidos por ASG particulares, pueden configurarse de manera tal que satisfagan una amplia gama de exigencias operacionales. Es posible configurar alguno de los ASG de modo que contengan aquellos asientos que representan dominios de denominación de nivel superior en alguna(s) parte(s) lógica(s) de la BIG, por ejemplo, la estructura organizacional de una gran compañía, pero no necesariamente todos los asientos subordinados. Otra posibilidad es configurar los ASG de tal modo que contengan sólo los contextos de denominación que representan principalmente asientos de hoja.

De las definiciones anteriores se sigue que los casos límite para un contexto de denominación pueden ser o bien un solo asiento o la totalidad del AIG.

Si bien la correspondencia lógica-física del AIG con los ASG es potencialmente arbitraria, la tarea de localización y gestión de la información se simplifica si los ASG se configuran de manera que contengan un pequeño número de contextos de denominación.

Para poder iniciar el tratamiento de una petición, un AUG debe contener alguna información en concreto, la dirección de presentación de por lo menos un ASG que él puede contactar inicialmente. La manera de adquirir y retener esa información es un asunto local.

Durante el proceso de modificación de asientos, la guía puede volverse incoherente. Es probable que ocurra esto, sobre todo, si la modificación afecta a alias o a objetos con alias que se hallen en diferentes ASG. Puede corregirse la incoherencia mediante acción específica del administrador; por ejemplo, eliminando alias si sus correspondientes objetos han sido eliminados. La guía seguirá funcionando durante este periodo de incoherencia.

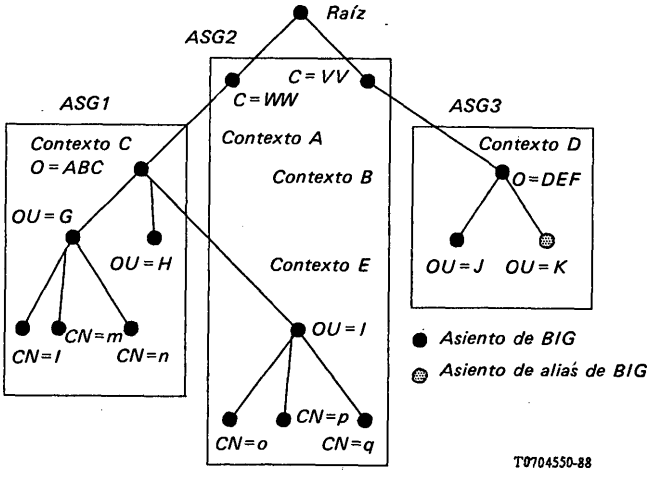



FIGURA 5/X.518

**AIG hipotético**

*Nota* - La raíz no está contenida en ningún ASG, pero debe existir alguna indicación a nivel local para distinguir los vértices (por ejemplo, C = VV, C = WW) que son subordinados inmediatos de la raíz.

## 10 Conocimiento

La BIG está potencialmente distribuida a través de múltiples ASG, conteniendo cada ASG un fragmento de BIG. Los principios que rigen la distribución de la BIG se especifican en el § 9 de la presente Recomendación.

Es exigencia de la guía el que, para modos particulares de interacción de los usuarios, su distribución se haga transparente, con lo que la BIG, en su conjunto, aparece como si estuviera en todos y cada uno de los ASG.

Para facilitar el cumplimiento de las exigencias operativas antes descritas, es preciso que cada ASG que contiene un fragmento de BIG sea capaz de identificar otros fragmentos de BIG contenidos en otros ASG y facultativamente, establecer con ellos una interacción.

En este punto se define el conocimiento como la base para la correspondencia de un nombre con su ubicación en un fragmento del AIG.

Los ASG contienen conceptualmente dos tipos de información:

a) información de guía;

b) información de conocimiento.

La *información de guía* está constituida por el conjunto de asientos en los que figura(n) el (los) contexto(s) de denominación para el (los) que el administrador de un determinado ASG tiene autoridad administrativa.

La *información de conocimiento* incorpora el (los) contexto(s) de denominación contenido(s) en un determinado ASG e indica cómo encaja(n) en la jerarquía global del AIG. La resolución de nombre, proceso por el que, dado el nombre de un asiento, se localiza el ASG que tiene autoridad administrativa sobre el mismo, se basa en la información de conocimiento.

Un *prefijo de contexto* es la secuencia de NDR que va de la raíz del AIG al vértice inicial de un contexto de denominación y corresponde al nombre distinguido de ese vértice.

Un *contexto de denominación* consta de una serie de referencias de conocimiento y de un prefijo de contexto. El contexto de denominación debe contener exactamente las siguientes referencias de conocimiento:

- Todas las referencias internas que definen la estructura interna de la porción del AIG incluida en el contexto de denominación.
- Todas las referencias subordinadas y subordinadas no específicas, a otros contextos de denominación.

### 10.1 *Referencias de conocimiento mínimo*

La guía tiene la propiedad de que pueda accederse a cada asiento con independencia de dónde se genere la petición.

Para conseguirlo, cada ASG mantendrá como mínimo las siguientes referencias de conocimiento:

- *referencias subordinadas* como se definen en el § 10.3.2 y/o referencias subordinadas no específicas, como se definen en el § 10.3.5; y
- *referencias superiores* como se definen en el § 10.3.3.

Así es posible establecer un *trayecto de referencia*, como una secuencia continua de referencias de conocimiento a todos los contextos de denominación de la guía.

Optativamente, para optimizar la calidad de funcionamiento, las *referencias recíprocas*, definidas en el § 10.3.4, pueden formar parte de un trayecto de referencia.

### 10.2 *Contexto de raíz*

Debido a la autonomía de los diferentes países u organizaciones globales, no hay probablemente un "único" ASG que contenga el contexto de raíz. La funcionalidad de un "ASG-raíz", por lo que se refiere al proceso de resolución de nombres, ha de ser proporcionada por los ASG que tienen autoridad administrativa para los contextos de denominación que son inmediatamente subordinados a la raíz. Estos ASG se denominan *ASG de primer nivel*. Cada ASG de primer nivel debe poder simular la

funcionalidad del "ASG-raíz". Para ello hace falta un conocimiento total sobre el contexto de denominación de raíz. El contexto de raíz es replicado sobre cada ASG de primer nivel y ha de ser, por tanto, administrado en común por las autoridades administrativas de primer nivel. Los procedimientos de administración deben establecerse mediante acuerdos multilaterales que quedan fuera del alcance de esta Recomendación.

- Cada ASG de primer nivel contendrá el contexto de raíz, lo que implica un trayecto de referencia a cada uno de los otros ASG de primer nivel.
- Cada ASG no-de-primer-nivel tendrá una referencia superior, lo que implica un trayecto de referencia a un ASG cualquiera de primer nivel.

10.3 *Referencias de conocimiento*

El conocimiento que posee un ASG se expresa en términos de un conjunto de una o más referencias de conocimiento, donde cada referencia asocia, directa o indirectamente, asientos de la BIG con los ASG que los contiene.

Para poder cumplir con el requisito de que cualquier asiento de la BIG pueda alcanzarse desde cualquier ASG, cada uno de éstos debe tener conocimiento de los asientos que el mismo contiene, y de los subordinados y posiblemente de los superiores de los mismos. Esto da lugar a los siguientes tipos de referencias de conocimiento:

- Referencias internas.
- Referencias subordinadas.
- Referencias superiores.
- Referencia subordinada no específica.

Se definen además, a efectos de optimización, los tipos de referencias facultativas que a continuación se indican:

- Referencias recíprocas.

Cuando en el conjunto de referencias de conocimiento asociado a un determinado ASG figuran sólo referencias internas, el ASG desconoce otros ASG y la BIG está, por tanto, centralizada.

10.3.1 *Referencias internas*

Una *referencia interna* consta de:

- el NDR correspondiente a un asiento de BIG;
- un puntero interno que señala dónde está almacenado el asiento en la BIG local. (La especificación del puntero queda fuera del alcance de esta Recomendación.)

Todos los asientos para los que un determinado ASG tiene autoridad administrativa están representados por referencias internas en la información de conocimiento de ese ASG.

10.3.2 *Referencias subordinadas*

Una *referencia subordinada* consta de:

- un NDR correspondiente a un asiento de BIG subordinado inmediato;
- el punto de acceso del ASG al que se delegó autoridad administrativa para ese asiento.

Todos los asientos subordinados contenidos en otro ASG y para los que dicho ASG ha delegado la autoridad administrativa deben estar representadas por referencias subordinadas (o referencias subordinadas no específicas como se indica en el § 10.3.5).

10.3.3 *Referencias superiores*

Una *referencia superior* consta de:

- punto de acceso de un ASG.

Cada ASG no-de-primer-nivel mantiene de forma precisa una referencia superior. La referencia superior formará parte de un trayecto de referencia hacia la raíz. Salvo que se utilice algún método ajeno al normalizado para conseguirlo, por ejemplo en un DGG, esto se llevará a cabo por referencia a un ASG que retiene un contexto de denominación cuyo prefijo de contexto tiene menos NDR que el prefijo de contexto que tiene menos NDR, retenido por ese ASG.

Si se presenta un nuevo ASG no-de-primer-nivel, éste debe tener un conocimiento inicial mínimo representado por la referencia superior. Cualquier conocimiento ulterior se agregará mediante

referencias subordinadas o recíprocas (tal como se describe en el § 10.3.4). Si se introduce un nuevo ASG de primer nivel, deberá obtener el contexto de raíz y comunicárselo a todos los demás ASG de primer nivel. La manera según la cual se lleva esto a cabo queda fuera del alcance de la presente Recomendación.

10.3.4 *Referencias recíprocas*

Una *referencia recíproca* consta de:

- un prefijo de contexto;
- el punto de acceso de un ASG que tiene autoridad administrativa para ese contexto de denominación.

Este tipo de referencia es facultativo y sirve para optimizar la resolución de nombre. Un ASG puede contener un número cualquiera (cero incluido) de referencias recíprocas.

10.3.5 *Referencias subordinadas no específicas*

Una *referencia subordinada no específica* consta de:

- el punto de acceso de un ASG que contiene uno o más contextos de denominación subordinados inmediatos.

Este tipo de referencia es facultativo. Con él se prevé el caso en que se tiene conocimiento de que un ASG contiene varios asientos subordinados pero se ignoran los NDR específicos de esos asientos.

Por cada contexto de denominación que contenga, un ASG puede contener un número cualquiera (cero incluido) de referencias subordinadas no específicas que será evaluado si se ha procedido al seguimiento de todas las referencias específicas, internas y subordinadas. Los ASG a los que se accede a través de una referencia no específica han de poder responder a las solicitudes directamente (éxito o fracaso). En caso de fracaso, se retorna al solicitante un **error de servicio** (serviceError) comunicándole un problema de **incapaz de proseguir** (unableToProceed).

10.4 *Administración del conocimiento*

Para operar una guía ampliamente distribuida, con un grado aceptable de coherencia y buen funcionamiento, hacen falta procedimientos que mantengan y amplíen el conocimiento contenido en cada ASG. Esos mismos procedimientos sirven para crear el conocimiento inicial.

El conocimiento puede mantenerse mediante:

a) La propagación, por parte del ASG o de su autoridad administrativa, de cambios de conocimiento a los ASG que contengan toda clase de referencias a aquél, cuando los cambios en el mismo invaliden las referencias. Esta es la única manera de mantener las referencias superiores, subordinadas y subordinadas no específicas.

b) La petición y obtención, por parte de los ASG, de referencias recíprocas para mejorar el funcionamiento mediante operaciones de guía ordinarias.

En esta Recomendación no se define ningún procedimiento para la propagación de los cambios de conocimiento descritos en a). Para ello deben establecerse localmente acuerdos bilaterales.

10.4.1 Petición de referencias recíprocas

Para mejorar el funcionamiento del sistema de guía, puede ampliarse el conjunto local de referencias recíprocas mediante operaciones de guía corrientes. Si un ASG tiene un puerto concatenado, puede solicitar a otro ASG (que debe tener también un puerto encadenado) que le remita las referencias de conocimiento que contienen información sobre la ubicación de contextos de denominación relacionados con el nombre de objeto deseado de una operación de guía ordinaria.

Si el componente **returnCrossReference** del **chainingArgument** se fija a TRUE, el componente **crossReference** del **chainingResult** puede estar presente y estará formado por una secuencia de ítems de referencia recíproca.

Cuando un ASG no es capaz de concatenar una petición al siguiente ASG, retornará un referimiento al ASG de origen. Si el componente **returnCrossReference** del argumento de concatenación era **TRUE**, el referimiento podrá contener, adicionalmente, el prefijo de contexto del contexto de denominación al cual él refiere. El componente **contextPrefix** estará ausente si el referimiento se basa en una referencia subordinada no específica. La referencia recíproca retornada por un referimiento sólo se basa en el conocimiento contenido en el ASG que generó el referimiento.

En ambos casos (resultado de concatenación y referimiento), una autoridad administrativa puede también optar, a través de su ASG, por ignorar la petición de retornar referencias recíprocas.

10.4.2 *Inconsistencias del conocimiento*

La guía ha de soportar mecanismos de verificación de la consistencia (coherencia) para garantizar un cierto grado de consistencia del conocimiento.

10.4.2.1 *Detección de inconsistencias del conocimiento*

El tipo de inconsistencia y la manera de detectarla varían de unos tipos a otros de referencia de conocimiento.

- *Referencias recíprocas y subordinadas:*

Esta clase de referencia no es válida si el ASG referenciado no tiene un contexto de denominación local con el prefijo de contexto contenido en la referencia. La inconsistencia se detecta al determinar el contexto de denominación inicial del proceso de resolución de nombre por los componentes progresión de operación y tipo de referencia del **ChainingArgument**.

- *Referencias subordinadas no específicas:*

Este tipo de referencia no es válido si el ASG referenciado no tiene un contexto de denominación local cuyo prefijo de contexto inmediatamente superior esté contenido en la referencia, es decir, la referencia contiene el contexto local del ASG menos el último NDR. Se aplica la prueba de coherencia como antes.

- *Referencias superiores:*

Una referencia superior no válida es una que no forma parte de un trayecto de referencia hacia la raíz. El mantenimiento de referencias superiores debe hacerse por medios externos y está fuera del ámbito de esta Recomendación.

*Nota* - No siempre es posible detectar una referencia superior no válida.

10.4.2.2 *Informes de inconsistencias de conocimiento*

Si se emplea la concatenación al efectuar una investigación de guía, todas las inconstancias serán detectadas por el ASG que contenga la referencia de conocimiento no válida al recibir un **serviceError** con problema de **invalidReference**.

Cuando un ASG retorna un referimiento basado en una referencia de conocimiento no válida, se remite al solicitante un **serviceError** con problema de **invalidReference** si utiliza el referimiento. La manera de comunicar la condición de error al ASG que almacena la referencia no válida esta fuera del ámbito de la presente Recomendación.

10.4.2.3 *Tratamiento de las referencias de conocimiento inconsistentes*

Después de detectar una referencia no válida, el ASG deberá tratar de restablecer la consistencia del conocimiento. Por ejemplo, esto puede hacerse eliminando simplemente una referencia recíproca no válida o reemplazándola por una correcta que puede obtenerse utilizando los mecanismos de **requestCrossReferences**.

El tratamiento que finalmente aplique un ASG a las referencias no válidas es una cuestión local que está fuera del ámbito de la presente Recomendación.

## SECCION 4 - *Servicio abstracto de ASG*

## 11 Visión de conjunto del servicio abstracto de ASG

11.1 El servicio abstracto de la guía se describe en forma completa en la Recomendación X.511. Cuando dicho servicio se proporciona en un entorno distribuido, tal como el modelado en el § 7 de la presente Recomendación, puede considerarse que es proporcionado por un conjunto de ASG. Esto es lo que se ilustra en la figura 1/X.518.

11.2 Para describir este modelo, cabe expresar el refinamiento del objeto de **guía** en sus objetos de **dsa** componentes como a continuación se indica:

```
DirectoryRefinement ::= REFINE directory AS
    dsa          RECURRING
        readPort              [S]  VISIBLE
        searchPort            [S]  VISIBLE
        modifyPort            [S]  VISIBLE
        chainedReadPort       PAIRED with dsa
        chainedSearchPort     PAIRED with dsa
        chainedModifyPort     PAIRED with dsa
```

11.3 El propio objeto **dsa** (ASG) puede definirse de la siguiente manera:

```
dsa OBJECT
    PORTS {  readPort              [S],
             searchPort            [S],
             modifyPort            [S],
             chainedReadPort,
             chainedSearchPort,
             chainedModifyPort}
::=  id-ot-dsa
```

El ASG suministra los puertos leer, buscar y modificar, haciendo así visibles esos servicios a los usuarios del objeto de guía, es decir, a los AUG. Además, un ASG soporta versiones "concatenadas" ("chained") de estos puertos, a saber: leer concatenado, buscar concatenado y modificar concatenado, lo que permite a los ASG propagar a otros ASG las peticiones de esos servicios.

11.4 Los puertos mencionados en los § 11.2 y 11.3 (excepto los definidos en la Recomendación X.511) se definen de la siguiente manera:

```
chainedReadPort     PORT
    ABSTRACT OPERATIONS {
        ChainedRead, ChainedCompare,
        ChainedAbandon}
    ::=  id-pt-chained-read

chainedSearchPort   PORT
    ABSTRACT OPERATIONS {
        ChainedList, ChainedSearch}
    ::=  id-pt-chained-search

chainedModifyPort   PORT
    ABSTRACT OPERATIONS {
        ChainedAddEntry,
        ChainedRemoveEntry,
        ChainedModifyEntry,
        ChainedModifyRDN}
    ::=  id-pt-chained-modify
```

## 12 Tipos de información

### 12.1 *Introducción*

12.1.1 En este punto se identifica, y en algunos casos se define, un cierto número de tipos de información que a continuación se emplean para definir varias de las operaciones del servicio abstracto de ASG. Los tipos de información de que se trata son los comunes a más de una operación; los que probablemente estarán vigentes en el futuro o son lo bastante complejos o autónomos como para merecer que se les defina con independencia de las operaciones que los utilizan.

12.1.2 Varios de los tipos de información empleados en la definición del servicio abstracto de ASG se definen de hecho, en otro sitio. En el § 12.2 se identifican estos tipos y se indica el origen de sus definiciones. En cada uno de los restantes puntos (§ 12.3 a 12.9) se identifica y define un tipo de información.

### 12.2 *Tipos de información definidos en otro lugar*

12.2.1 En la Recomendación X.501 se definen los siguientes tipos de información:

a) **aliasedObjectName**;

b) **DistinguishedName;**

c) **Name;**

d) **RelativeDistinguishedName.**

12.2.2 En la Recomendación X.511 se definen los siguientes tipos de información:

(Vincular-abstracto)

a) **DirectoryBind;**

(Operaciones-abstractas)

b) **Abandon;**

(Errores-abstractos)

c) **Abandoned;**

d) **AttributeError;**

e) **NameError;**

f) **SecurityError;**

g) **ServiceError;**

h) **UpdateError;**

(Macro)

i) **OPTIONALLY-SIGNED.**

(Data Type)

j) **SecurityParameters.**

12.2.3 En la Recomendación X.520 se define el siguiente tipo de información:

a) **PresentationAddress**

12.3 *Argumentos de concatenación*

12.3.1 Los **ChainingArguments** están presentes en toda operación abstracta concatenada, transportando a un ASG la información que necesita para llevar a cabo correctamente su parte de la tarea global:

```
ChainingArguments ::=SET {
    originator          [0]  DistinguishedName OPTIONAL,
    targetObject        [1]  DistinguishedName OPTIONAL,
    operationProgress   [2]  OperationProgress DEFAULT {notStarted},
    traceInformation    [3]  TraceInformation,
    aliasDereferenced   [4]  BOOLEAN DEFAULT FALSE,
    aliasedRDNs         [5]  INTEGER OPTIONAL,
        -- ausente a menos que aliasDereferenced sea TRUE --
    returnCrossRefs     [6]  BOOLEAN DEFAULT FALSE,
    referenceType       [7]  ReferenceType DEFAULT superior,
    Info                [8]  DomainInfo OPTIONAL,
    timeLimit           [9]  UTCTime OPTIONAL,
                        [10] SecurityParameters DEFAULT {}}
```

12.3.2 Los diversos componentes tienen el significado que se define en los § 12.3.2.1 a 12.3.2.11.

12.3.2.1 El componente **originador** transporta el nombre del (último) originador de la petición, a menos que haya sido ya especificado en los parámetros de seguridad. Si el solicitante está presente en **CommonArguments**, este argumento puede omitirse.

12.3.2.2 El componente **targetObject** lleva el nombre del objeto a cuyo asiento de guía se encamina. El cometido de este objeto depende de la operación-abstracta de que se trate en concreto: puede ser el objeto sobre cuyo asiento se va a actuar o que va a ser objeto base para una petición o subpetición en la que participen múltiples objetos (por ejemplo, **ChainedList** o **ChainedSearch**). Este componente sólo podrá omitirse si hubiera tenido algún valor como el parámetro de objeto de base en **XArgument** (véase el § 14.3.1), en cuyo caso su valor implícito será ese valor.

12.3.2.3 El componente **operationProgress** se utiliza para informar al ASG sobre el avance de la operación, y por ende del papel que se espera que desempeñe en la realización global de la misma. La información transportada en este componente se especifica en el § 12.5.

12.3.2.4 El componente **traceInformation** se emplea para impedir la formación de bucles entre los ASG cuando está actuando la concatenación. Un ASG añade un nuevo elemento a la información de rastreo antes de concatenar una operación a otro ASG. Cuando se pide a un ASG que efectúe una operación, éste comprueba que la operación no ha formado un bucle examinando la información de rastreo. La información transportada en este componente se especifica en el § 12.6.

12.3.2.5 El componente **aliasDereferenced** es un valor booleano que se usa para indicar si, en cada momento y en el transcurso de la resolución de nombre distribuido, se han encontrado y desreferenciado, o no, asientos de alias. El valor por defecto de **FALSE** indica que no se ha desreferenciado ningún asiento de alias.

12.3.2.6 El componente **aliasedRDNs** indica cuántos de los NDR en el **targetObjectName** se han generado a partir de los atributos **aliasedObjectName** de uno (o más) asientos de alias. El valor entero se fija cada vez que un asiento de alias es encontrado y desreferenciado. Este componente estará presente si, y únicamente si, el componente **aliasDereferenced** es **TRUE**.

12.3.2.7 El componente **returnCrossRefs** es un valor booleano que indica si se pide, o no, que las referencias de conocimiento utilizadas al efectuar una operación distribuida sean devueltas al ASG inicial como referencias recíprocas, junto con un resultado o un referimiento. El valor por defecto de **FALSE** indica que dichas referencias de conocimientos no deben ser retornadas.

12.3.2.8 El componente **referenceType** indica al ASG al que se pide que realice la operación-abstracta, qué tipo de conocimiento se empleó para encaminarle la petición. El ASG puede así ser capaz de detectar errores en el conocimiento contenido por el invocante. Si se detecta un error, se indicará mediante un **ServiceError** con el problema **InvalidReference**. En el § 12.7 se define **ReferenceType**.

*Nota* - Si no está el **referenceType** se supondrá que el valor es **superior**.

12.3.2.9 El componente **info** se emplea para transportar información específica del DGG entre los ASG que participan en el tratamiento de una petición común. Este componente es del tipo **Domaininfo**, que es un tipo no restringido:

**Domaininfo ::= ANY**

12.3.2.10 El componente **timeLimit**, si está presente, indica el momento en que la operación debe estar concluida.

12.3.2.11 El componente **SecurityParameters** se especifica en la Recomendación X.511. Su ausencia se considera equivalente a que haya un conjunto vacío de parámetros de seguridad.

12.4 *Resultados de la concatenación*

12.4.1 Los **ChainingResults** están presentes en el resultado de cada operación-abstracta, y proporcionan una realimentación al ASG que la invocó.

```
ChainingResults ::=    SET {
    Info               [0] DomainInfo  OPTIONAL,
    crossReferences    [1] SEQUENCE OF CrossReference OPTIONAL}
                       [2] SecurityParameters DEFAULT {}}
```

12.4.2 Los distintos componentes tienen los significados que se definen en los § 12.4.2.1 y 12.4.2.3.

12.4.2.1 El componente **info** se emplea para transportar información específica del DGG entre los ASG que participan en el tratamiento de una petición común. Este componente es del tipo **Domaininfo**, que es un tipo no restringido.

12.4.2.2 El componente **crossReferences** no está presente en los **ChainingResults** a menos que el componente **returnCrossRefs** de la correspondiente petición tenga el valor **TRUE**. Este componente consta de una secuencia de ítems de **CrossReference** cada uno de los cuales contiene un **contextPrefix** y un descriptor de **accessPoint** (véase el § 12.8).

```
CrossReference      ::=  SET{
    contextPrefix   [0]  DistinguishedName,
    accessPoint     [1]  AccessPoint}
```

Un ASG puede añadir una **crossReference** cuando hace concordar una parte del argumento del **targetObject** de una operación-abstracta con uno de sus prefijos de contexto. Es posible que la autoridad administrativa de un ASG siga la política de no retornar ese conocimiento, por lo que, en este caso, no añadirá un ítem a la secuencia.

12.4.2.3 El componente **SecurityParameters** se especifica en la Recomendación X.511. Su ausencia se considera equivalente a que haya un conjunto vacío de parámetros de seguridad.

12.5 *Avance de la operación*

12.5.1 El valor **OperationProgress** describe el grado de avance en la realización de una operación abstracta en la que deben participar varios ASG.

```
OperationProgress ::=      SET {
    nameResolutionPhase    [0]
        ENUMERATED {
            notStarted     (1),
            proceeding     (2),
            completed      (3)},
    nextRDNToBeResolved    [1]
            INTEGER OPTIONAL}
```

12.5.2 Los distintos componentes tienen los significados que se definen en los §12.5.2.1 y 12.5.2.2.

12.5.2.1 El componente **nameResolutionPhase** indica qué fase se ha alcanzado en el tratamiento del nombre del **targetObject** de una operación. Cuando indica que la resolución de nombre sigue estando **notStarted** quiere decir que aún no se ha alcanzado un ASG con un contenido de denominación que contenga el o los NDR iniciales del nombre. Si la resolución de nombres está **proceeding**, se ha reconocido la parte inicial del nombre, si bien todavía no se ha alcanzado el ASG que contiene el objeto buscado. El componente **nextRDNoBeResolved** indica qué proporción del nombre ha sido reconocida (en el § 12.5.2.2). Si la resolución del nombre está **completed**, quiere decirse que se ha alcanzado el ASG que contiene el objeto buscado y se está procediendo a la realización de la operación propiamente dicha.

12.5.2.2 El componente **nextRDNToBeResolved** indica al ASG cuál de los NDR que figuran en el nombre del **targetObject** es el próximo a resolver. Adopta la forma de un número entero de la gama que va de uno al número de NDR del nombre. Este componente sólo está presente si el **nameResolutionPhase** tiene el valor **proceeding**.

12.6 *Información de rastreo*

12.6.1 El valor **TraceInformation** lleva un registro de los ASG que han participado en la realización de una operación. Se utiliza para detectar, o evitar, la existencia de bucles que podrían formarse debido a conocimientos incoherentes o a la presencia de bucles de alias en el AIG.

```
TraceInformation    ::=  SEQUENCE OF TraceItem
TraceItem           ::=  SET {
    dsa                  [0]  Name,
    targetObject         [1]  Name OPTIONAL,
    operationProgress    [2]  OperationProgress }
```

12.6.2 Todo ASG que propaga una operación a otro ASG, añade un nuevo ítem a la información de rastreo. Cada uno de esos **TraceItem** contiene:

a) el **Name** del ASG que agrega el ítem;

b) el nombre **targetObjectName** que fue recibido, en la respuesta entrante, por el ASG que añade el ítem. Este parámetro se admite si la consulta (o indagación) que se está concatenando procede de un AUG (en cuyo caso el valor implícito es el **object** o el **baseObject** en la **XOperation**), o si su valor es el mismo del **targetObject** (real o implícito) en el **ChainingArgument** de la petición saliente;

c) el **operationProgress** recibido, en la petición entrante, por el ASG que está añadiendo el ítem.

12.7 *Tipo de referencia*

12.7.1 Un valor **ReferenceType** indica una de las diversas clases de referencias que se definen en el § 10.

```
ReferenceType ::=
    ENUMERATED {
        superior                (1),
        subordinate             (2),
        cross                   (3),
        nonSpecificSubordinate  (4)}
```

12.8 *Punto de acceso*

12.8.1 Un valor **AccessPoint** identifica un determinado punto en el que se puede producir acceso a la guía, y más concretamente a un ASG. El punto de acceso tiene un nombre, el del ASG afectado, y una **PresentationAddress** a utilizar en las comunicaciones de la ISA con ese ASG.

```
AccessPoint     ::=  SET {
    ae-title    [0]  Name,
    address     [1]  PresentationAddress }
```

12.9 *Referencia de continuación*

12.9.1 Una **ContinuationReference** describe cómo continuar la realización de toda o parte de una operación-abstracta en uno o unos ASG diferentes. Se retorna, típicamente, a modo de referimiento cuando el ASG implicado no puede o no quiere propagar la petición por sí mismo.

```
ContinuationReference       ::= SET {
    targetObject        [0]  Name,
    aliasedRDNs         [1]  INTEGER OPTIONAL,
    operationProgress   [2]  OperationProgress,
    rdnsResolved        [3]  INTEGER OPTIONAL,
    referenceType       [4]  ReferenceType OPTIONAL,

    -- sólo presente en el PSG
    accessPoints        [5]  SET OF AccessPoint}
```

12.9.2 Los distintos componentes tienen los significados que se definen en los § 12.9.2.1 a 12.9.2.6.

12.9.2.1 El **targetObject Name** cuya utilización, en la continuación de la operación, se propone. Este podría ser diferente del **targetObject Name** recibido en la petición entrante si, por ejemplo, se ha desreferenciado un alias, o si se ha localizado el objeto de base en una búsqueda.

12.9.2.2 El componente **aliasedRDNs** indica cuántos de los NDR (en su caso), del nombre deseado, han sido producidos desreferenciando un alias. El argumento sólo está presente si se ha desreferenciado un alias.

12.9.2.3 El **operationProgress** alcanzado, que regirá la ulterior realización de la operación-abstracta por los ASG denominados, en caso de que el ASG o AUG que reciba la **ContinuationReference**, la siga.

12.9.2.4 El valor del componente **rdnsResolved** (que sólo debe estar presente si algunos de los NDR del nombre no han sido sometidos a resolución de nombres total, pero se ha supuesto que son correctos, como consecuencia de una referencia recíproca) indica cuántos NDR se han resuelto efectivamente utilizando referencias internas únicamente.

12.9.2.5 El componente **referenceType**, que sólo figura en el servicio abstracto de ASG, indica qué tipo de conocimiento se utilizó para generar esta continuación.

12.9.2.6 El componente **accessPoints** indica los puntos de acceso que deben ser seguidos para lograr esa continuación. Cuando en el proceso participan referencias subordinadas no específicas puede haber listado más de un **AccessPoint** debiendo seguirse cada uno de ellos, por ejemplo, por difusión.

## 13 Vincular-abstracto y desvincular-abstracto

Un ASG utiliza una **DSABind** y una **DSAUnbind** al comienzo y al final, respectivamente, de un periodo de acceso a otro ASG.

13.1 *Vinculación de ASG*

13.1.1 Un ASG utiliza una operación-vincular-abstracta **DSABind** para vincular sus puertos **chainedRead**, **chainedSearch** y **chainedModify** a los de otro ASG.

```
DSABind ::=             ABSTRACT-BIND
    TO                  {chainedRead,
                         chainedSearch,
                         chainedModify}
    DirectoryBind
```

13.1.2 Los componentes de la **DSABind** son idénticos a sus contrapartes en el **DirectoryBind** (véase la Recomendación X.511), con las siguientes diferencias.

13.1.2.1 Las **Credentials** del **DirectoryBindArgument** permiten enviar al ASG que responde una información que identifica el título EA del ASG iniciador. El título EA tiene que ser de la forma de un Nombre Distinguido de Guía.

13.1.2.2 Las **Credentials** del **DirectoryBindResult** permite enviar al ASG iniciador información que identifica el título EA del ASG que responde. El título EA tendrá la forma de un Nombre Distinguido.

13.2 *Desvinculación de ASG*

13.2.1 Se emplea la operación **DSAUnbind** para desvincular los puertos leer concatenado, buscar concatenado y modificar concatenado de un par de ASG.

```
DSAUnbind    ::=    ABSTRACT-UNBIND
    FROM            {chainedRead,
                     chainedSearch,
                     chainedModify}
```

13.2.2 No hay argumentos ni resultados ni errores.

## 14 Operaciones-abstractas concatenadas

14.1 En correspondencia con cada uno de los puertos del servicio abstracto de guía hay un puerto de ASG, lo que permite que el servicio abstracto sea prestado por ASG cooperantes. Las operaciones abstractas en los puertos correspondientes están también en correspondencia biunívoca. Los nombres de los puertos y de las operaciones-abstractas se han elegido de tal manera que reflejen esa correspondencia, estando formados los de los puertos u operaciones-abstractas del servicio abstracto de ASG por los correspondientes del servicio abstracto de guía a los que se ha antepuesto la palabra "chained". Los puertos y operaciones-abstractas resultantes son los siguientes:

**ChainedReadPort:** **ChainedRead,**
**ChainedCompare,**
**ChainedAbandon**

**ChainedSearchPort:** **ChainedList,**
**ChainedSearch**

**ChainedModifyPort:** **ChainedAddEntry,**
**ChainedRemoveEntry,**
**ChainedModifyEntry,**
**ChainedModifyRDN**

14.2 Los argumentos, resultados y errores de las operaciones-abstractas concatenadas se forman, salvo en un caso, de forma sistemática, a partir de los argumentos, resultados y errores de las correspondientes operaciones-abstractas del servicio abstracto de guía (tal como se describe en el § 14.3). La excepción es la operación-abstracta **ChainedAbandon**, que es sintácticamente equivalente a su contrapartida del servicio-abstracto de guía (descrita en el § 14.4).

14.3 Una operación-abstracta **ChainedX** se emplea para propagar entre los ASG una petición que se originó (normalmente) por la invocación, por parte de un AUG, de una operación-abstracta **X** en un ASG que eligió concatenarla. Los argumentos de la operación-abstracta pueden ser firmados, facultativamente, por el invocador y, si así se pide, el ASG que lleva a cabo la operación puede firmar los resultados.

14.3.1 La derivación sistemática de una operación-abstracta concatenada **ChainedX** a partir de su contraparte **X** es como sigue:

dado:

```
X    ::=
     ABSTRACT-OPERATION
          ARGUMENT    XArgument
          RESULT      XResult

          ERRORS {..., Referral,...}
```

la operación-abstracta concatenada se deriva así:

```
ChainedX ::=
    ABSTRACT-OPERATION
        ARGUMENT  OPTIONALLY-SIGNED SET{
                  ChainingArgument,
                  [0]  XArgument}
    RESULT  OPTIONALLY-SIGNED SET{
                  ChainingResult,
                  [0] XResult}
    ERRORS    {...,DsaReferral,...}
```

*Nota* - La especificación definitiva del servicio abstracto de ASG en el Anexo A, aplica totalmente esta deducción a las operaciones-abstractas concatenadas.

14.3.2 Los argumentos de la operación-abstracta derivada tienen los significados que se describen en los § 14.3.2.1 y 14.3.2.2.

14.3.2.1 El **ChainingArgument** contiene la información, a propósito y más allá de los argumentos originales suministrados por el AUG que se necesita para que el ASG actuante lleve a cabo la operación. Este tipo de información se define en el § 12.3.

14.3.2.2 El **XArgument** contiene los argumentos originales suministrados por el AUG tal como se especifica en la cláusula correspondiente de la Recomendación X.511.

14.3.3 Si la petición tiene éxito, se retorna el resultado. Los parámetros del resultado tienen el significado que se describe en los § 14.3.3.1 y 14.3.3.2.

14.3.3.1 El **ChainingResult** contiene la información a propósito y más allá de la que se ha de suministrar al AUG de origen, que podrían necesitar varios ASG anteriores de una cadena. Este tipo de información se define en el § 12.4.

14.3.3.2 El **XResult** contiene el resultado que retorna el actuante en esta ordenación-abstracta y que se pretende devolver en el resultado al ASG de origen. Esta información es tal como se especifica en la cláusula correspondiente de la Recomendación X.511.

14.3.4 Si la petición fracasa, se retorna uno de los errores listados. El conjunto de errores que pueden ser comunicados es el descrito para la correspondiente operación-abstracta en la Recomendación X.511, excepto que se retorna **DSAReferral** en vez de **Referral**. Los distintos errores se definen o referencian en el § 15.

14.4 Un ASG usa una operación abstracta **ChainedAbandon** para indicar a otro ASG que ya no le interesa que se efectúe una operación concatenada, invocada previamente. Esto puede ocurrir por diversas razones, de las que son ejemplo las siguientes:

a) la propia operación que llevó originalmente a la concatenación del ASG ha sido abandonada, o ha sido abortada implícitamente por la ruptura de una asociación;

b) el ASG ha obtenido la información necesaria de otra manera, por ejemplo, de un ASG de respuesta más rápida que participa en una difusión.

Un ASG nunca está obligado a emitir un **ChainedAbandon**, ni de hecho a abandonar efectivamente una operación cuando se le pide que lo haga.

Si el **ChainedAbandon** logra efectivamente para la realización de una operación, se retornará un resultado y la operación supeditada retornará un error-abstracto de **Abandoned**. Si la **ChainedAbandon** no logra detener la operación, ella misma retornará un error de **AbandonFailed**.

## 15 Errores-abstractos concatenados

### 15.1 *Introducción*

15.1.1 En su mayor parte, pueden ser retornados en el servicio abstracto de ASG los mismos errores-abstractos que en el de guía. La excepción es que se retorne el "error" **DSAReferral** (véase el § 15.2) en vez de **Referral**; los problemas de servicio siguiente tienen la misma sintaxis abstracta pero diferentes semánticas:

a) **invalidReference**;

b) **loopDetected**.

15.1.2 La prioridad entre los errores abstractos que puedan producirse es la misma que para el servicio abstracto de guía, como se especifica en la Recomendación X.511.

15.2 *Referimiento de ASG*

15.2.1 El error-abstracto **DSAReferral** lo genera un ASG cuando, por la razón que sea, no desea continuar con la realización de una operación-abstracta por concatenación o difusión de la operación-abstracta a otro u otros ASG. Las circunstancias en las que puede retornar un referimiento se describen en el § 8.4.

```
DSAReferral ::=
    ABSTRACT-ERROR
        PARAMETER SET{
            [0]ContinuationReference,
            contextPrefix [1] DistinguishedName OPTIONAL }
```

15.2.2 Los distintos parámetros tienen los significados que se describen en los § 15.2.2.1 y 15.2.2.2.

15.2.2.1 La **ContinuationReference** contiene la información que necesita el invocador para propagar una petición ulterior adecuada, quizá a otro ASG. Este tipo de información se especifica en el § 12.9.

15.2.2.2 Si el componente **returnCrossRefs** de los **ChainingArguments**, para esta operación-abstracta, tiene el valor **TRUE**, y el referimiento se basa en una referencia subordinada o recíproca, se puede incluir facultativamente, el parámetro **contextPrefix**. La autoridad administrativa de cualquier ASG decidirá qué referencias de conocimiento pueden retornarse, si es que se retorna alguna, de esta manera (las otras podrían, por ejemplo, tener carácter confidencial para ese ASG).

## SECCION 5 - *Procedimientos de operaciones distribuidas*

### 16 Introducción

16.1 *Alcance y límites*

En este punto se especifican los procedimientos que llevan a cabo los ASG para el funcionamiento distribuido de la Guía. Cada ASG sigue individualmente los procedimientos que se indican más abajo; la acción colectiva de todos ellos produce el conjunto completo de servicios proporcionados a los usuarios por la guía.

La descripción de procedimientos para un único ASG se basa en los modelos de los § 7 a 10 de esta Recomendación.

Téngase en cuenta que el modelo y los procedimientos se incluyen a efectos expositivos solamente y que no se pretende con ellos limitar o dirigir la ejecución de un ASG real.

Este punto se divide en tres subpuntos: la presente introducción, un modelo conceptual para describir el comportamiento de la guía, y una introducción de los modelos basados en el ASG y en la operación propiamente dicha, de las operaciones del ASG.

16.2 *Modelo conceptual*

La complejidad de la operación distribuida de la guía plantea la necesidad de una modelación conceptual utilizando técnicas descriptivas, tanto narrativas como gráficas. No deberán interpretarse, no obstante, la narrativa y los diagramas gráficos como una descripción formal de la operación de guía distribuida.

16.3 *Operación individual y cooperativa de los ASG*

El modelo contempla el funcionamiento de los ASG desde dos perspectivas distintas que, en conjunto, dan una imagen operativa completa de la guía.

a) *Perspectiva centrada en el ASG.* En esta perspectiva, el conjunto de procedimientos que sustentan la guía se describen desde el punto de vista de un ASG individual. Ello hace posible proporcionar una especificación definitiva de cada procedimiento y tener totalmente en cuenta sus interrelaciones y su estructura de control global. En el § 18 se describen los procedimientos de ASG según una perspectiva centrada en el ASG.

b) *Perspectiva centrada en la operación.* La perspectiva centrada en el ASG da una imagen plenamente detallada pero hace difícil la comprensión de la estructura de las operaciones individuales, que pueden ser procesadas por múltiples ASG. Por ello, en el 17 se adopta una perspectiva centrada principalmente en las operaciones para ir presentando las fases de tratamiento aplicables a cada una.

En apoyo del funcionamiento distribuido de la guía, cada ASG debe llevar a cabo las acciones necesarias para comprender el propósito de cada operación y acciones adicionales para distribuir esa realización entre múltiples ASG. En el § 17 se analiza la distinción entre estas dos clases de acciones. En el § 18 se especifican en detalle ambas clases de acciones.

## 17 Comportamiento de guía distribuida

### 17.1 *Realización cooperativa de operaciones*

Cada ASG dispone de procedimientos que le permiten la realización total de todas las operaciones de guía. Si un ASG contiene íntegramente la BIG, de hecho, todas las operaciones se realizan dentro de ese ASG. Si la BIG está distribuida entre múltiples ASG, la ejecución de una operación típica se fragmenta, realizándose simplemente una porción de esa operación en cada uno de los ASG cooperantes, que pueden ser muchos.

En el entorno distribuido, el ASG típico ve cada operación como un suceso transitorio: la operación es invocada por un AUG o por algún otro ASG; el ASG procede al tratamiento del objeto y lo dirige a continuación hacia otro ASG para ulterior tratamiento.

Otra posible perspectiva considera el tratamiento total de que ha sido objeto una operación durante su realización por múltiples ASG cooperantes. En esta perspectiva se ponen de manifiesto las fases del tratamiento comunes a todas las operaciones.

### 17.2 *Fases del procesamiento de operaciones*

Cabe descomponer cada operación de guía en las tres distintas fases siguientes:

a) la fase de resolución de nombre, en la que el nombre del objeto, en cuyo asiento se va a efectuar una determinada operación, se utiliza para localizar el ASG que contiene el asiento;

b) la fase de evaluación, en la que la operación especificada por una peticón de guía particular (por ejemplo, leer) se lleva a cabo efectivamente;

c) la fase de fusión de resultados, en la que los resultados de una operación especificada se retorna al AUG solicitante. Si se eligió un modo de interacción por concatenación, la fase de fusión de resultados puede comprender varios ASG, cada uno de los cuales habría concatenado la petición o subpetición original (definida en el § 17.3.1 descomposición de la petición) a otro ASG durante alguna, o ambas, de las fases precedentes.

En el caso de las operaciones **Read**, **Compare**, **List**, **Search**, y **ModifyEntry**, la resolución de nombre tiene lugar sobre el nombre del objeto proporcionado en el argumento de la operación. En el caso de **AddEntry**, **RemoveEntry**, y **ModifyRDN**, la resolución de nombre tiene lugar sobre el nombre del objeto inmediatamente superior (obtenido retirando el NDR final del nombre proporcionado en el argumento de la operación).

Una operación sobre un determinado asiento puede dirigirse inicialmente a cualquier ASG de la guía. Ese ASG utiliza su conocimiento, posiblemente junto con otros ASG, para tratar la operación a lo largo de las tres fases.

#### 17.2.1 *Fase de resolución de nombre*

La resolución de nombre es un proceso en el que se trata de concordar de forma secuencial cada NDR de un nombre contemplado con un arco (o vértice) del AIG, comenzando lógicamente en la raíz y descendiendo por el AIG. Sin embargo, como el AIG está distribuido arbitrariamente entre muchos ASG, cada ASG sólo puede efectuar una fracción del proceso de resolución de nombre. Un determinado ASG efectúa su fracción del proceso investigando su conocimiento local. El procedimiento se describe en

el § 18.6 y en los correspondientes diagramas de las figuras 11/X.518 a 13/X.518. Cuando un ASG alcanza el límite de su contexto de denominación sabrá, del conocimiento en él contenido, si la resolución puede continuarla otro ASG o si el nombre es erróneo.

17.2.2 *Fase de evaluación*

Completada la fase de resolución de nombre, se realiza la operación (por ejemplo, leer o buscar) que de hecho se pretende.

Las operaciones que afectan a un solo asiento - **Read, Compare, AddEntry, RemoveEntry, ModifyRDN,** y **ModifyEntry** - pueden llevarse a cabo por entero en el ASG en que se ha localizado ese asiento. **AddEntry, RemoveEntry** y **Modify RDN** pueden afectar el conocimiento de más de un ASG. Véase el § 18.7.1.

Las operaciones que afectan a múltiples asientos - **List** y **search** - necesitan localizar subordinadas del objetivo que pueden hallarse o no en el mismo ASG. Si no están todos en el mismo ASG, han de dirigirse las operaciones hacia los ASG especificados en las referencias subordinadas para completar el proceso de evaluación.

17.2.3 *Fase de fusión de resultados*

Se entra en la fase de fusión de resultados cuando se dispone de algunos de los resultados de la fase de evaluación.

En los casos en que la operación ha afectado solamente a un asiento, puede retornarse simplemente el resultado de la operación al AUG solicitante. Cuando ha afectado a múltiples asientos en múltiples ASG, los resultados han de combinarse.

Las respuestas permitidas, retornadas al solicitante tras la fusión, son las siguientes:

a) un resultado completo de la operación;

b) un resultado que no es completo porque algunas partes del AIG permanecen inexploradas (sólo en los casos de **List** y **Search**). Ese *resultado parcial* puede contener referencias de continuación para las partes del AIG no exploradas;

c) un error (el referimiento es un caso especial).

d) y si el solicitante era un ASG, un ChainingResult.

17.3 *Gestión de operaciones distribuidas*

El argumento de cada operación-abstracta que se le puede pedir que efectúe a un ASG, contiene información que indica el avance de la operación a medida que transita por distintos ASG de la Guía. Esto permite que cada ASG realice el aspecto adecuado del tratamiento requerido y también registrar la terminación de ese aspecto antes de dirigir la operación hacia otros ASG.

En el ASG se incluyen procedimientos adicionales para la distribución física de las operaciones y para satisfacer otras necesidades que plantee esa distribución.

17.3.1 *Descomposición de petición*

La descomposición de la petición es un proceso que realiza internamente un ASG antes de comunicar con otro u otros ASG. Una petición se descompone en varias subpeticiones, de tal modo que cada uno de los otros ASG ejecute una parte de la tarea original. Se puede utilizar la descomposición de petición, por ejemplo, en la operación de buscar, después de haber encontrado el objeto base. Tras la descomposición, cada subpetición puede ser entonces concatenada o difundida a otros ASG que continúan la tarea.

17.3.2 *El ASG como respondedor de petición*

Un ASG que recibe una petición puede verificar el grado de avance de esa petición utilizando el parámetro avance de la operación. Así determinará si la operación está todavía en la fase de resolución de nombre o ha alcanzado la fase de evaluación y qué porción de la operación debería tratar de satisfacer el ASG. Si el ASG no puede satisfacer totalmente la petición, deberá pasarla a otro u otros ASG que puedan ayudarle a cumplimentar la petición (por concatenación o difusión) o retornar un referimiento a otro ASG o concluir la petición con un error.

17.3.3 *Compleción de operaciones*

Todo ASG que inicia una operación o la propague a otro u otros ASG, debe proceder a su seguimiento hasta que cada uno de los otros ASG haya retornado un resultado o un error, o haya transcurrido el tiempo máximo límite de la operación. Esta exigencia se aplica a todas las operaciones, modos de propagación y fases de tratamiento. Con ello se garantiza el cierre ordenado de operaciones distribuidas que se hubieran propagado y entrado en la guía.

17.4 *Otras consideraciones relativas a la operación distribuida*

17.4.1 *Validación de petición*

Al recibir una operación de guía, un ASG debe, en primer lugar, validar la operación para asegurarse de que puede avanzar. Circunstancias tales como bucles en el AIG originados por un empleo inadecuado de los alias o la utilización de conocimiento erróneo pueden dar lugar a que se envíen operaciones a ASG incapaces de tratarlas.

En un caso sencillo, estas circunstancias erróneas son tratadas adecuadamente por los procedimientos de resolución de nombres tal como se describe en el § 18. Sin embargo, cuando las circunstancias hacen que las operaciones formen bucles (como se describe en el § 17.4.3) no basta con la resolución de nombres únicamente.

Las acciones de validación de peticiones garantizan la detección de un bucle antes de que se intente hacer avanzar una operación debido a la presencia de datos erróneos originados por el bucle. El proceso de detección se realiza por el procedimiento de detección de bucles especificado en el § 18.5.1.

Cuando hay procedimientos de seguridad en vigor, la validación de petición verifica además la identidad del ASG o AUG peticionario y la validez de la petición.

17.4.2 *Información de estado y de rastreo*

El grado de avance de una operación en la guía y la presencia de condiciones de bucle se determinan mediante el "estado" de la operación, entendiéndose por estado lo siguiente:

- el nombre del ASG que en cada momento está procesando la operación.
- el nombre del **targetObject** que figura en el argumento de la operación.
- el **operationProgress** que figura en el argumento de la operación y que se define en el § 12.5.

Un ASG necesita conocer todos los estados anteriores de una operación además del actual. Esos estados se registran en el argumento **traceInformation** y se transportan con la operación.

El argumento **traceInformation** constituye la base de las estrategias de prevención/detección de bucles que se especifican en el § 17.4.3.

17.4.3 *Formación de bucles*

En el contexto de una operación de guía particular se produce un bucle si, en cualquier momento, la operación vuelve a un estado previo (según se ha definido más arriba). Para el tratamiento de los bucles se utiliza el argumento **traceInformation**. Se han definido dos estrategias para el tratamiento de bucles: la de detección y la de prevención de los mismos. En la *detección de bucle*, un ASG determina si se ha producido un bucle en una operación entrante y, si es así, retorna un error. En la *prevención de bucle*, el ASG determina sí, caso de proseguirse con una operación, se produciría un bucle.

17.4.4 *Controles de servicio*

Algunos controles de servicio requieren una consideración especial en el entorno distribuido para que la operación sea tratada del modo que se solicitó.

a) **chainingProhibited**: Un ASG consulta este control de servicio para determinar el modo de propagación de una operación. Si está establecido el control, el ASG utiliza siempre el modo referimiento. Pero si no lo está, puede elegir entre emplear concatenación o referimiento según sus capacidades.

b) **timeLimit**: Un ASG debe tener en cuenta este control de servicio para asegurarse de que no se rebasa el límite de tiempo. El ASG al que un AUG pide que realice una operación toma nota al principio del **timelimit**, expresado por el AUG como tiempo disponible transcurrido, en segundos, para la terminación de la operación. Si hace falta concatenación, se incluye el **timeLimit** en el argumento de concatenación que se pasa al (o a los) siguiente(s) ASG.

En este caso se emplea para cada petición concatenada el mismo valor de límite de tiempo que es el tiempo (UTC) en que la operación debe estar concluida para cumplir con la limitación especificada originalmente. Si recibe un argumento de concatenación con un **timeLimit** especificado, el ASG receptor respeta ese límite.

c) **sizeLimit**: Un ASG ha de respetar este control de servicio para garantizar que la lista de resultados no supera el tamaño especificado. El límite, tal como figura en el argumento común de la petición original, se transporta inalterado en las concatenaciones/difusiones de la petición. Si hace falta una descomposición de la petición, se incluye el mismo valor en el argumento que se pasa al siguiente ASG, esto es, se emplea el límite total en cada sub-petición. Cuando se retornan los resultados, el ASG solicitante desglosa los resultados múltiples y aplica el límite al total para asegurar que sólo se retorna el número pedido. Si se ha superado el límite, se indica en la respuesta.

d) **Priority**: Los ASG tienen la responsabilidad, cualquiera que sea el modo de propagación, de garantizar que en el tratamiento de las operaciones, mantiene el orden que satisface este control de servicio, si es que existe.

e) **localScope**: La operación está limitada a un alcance definido localmente y no puede propagarse por ninguno de los modos.

f) **scopeOfReferral**: Si el ASG devuelve un referimiento o un resultado parcial a una operación **List** o **Search**, las **ContinuationReferences** incluidas estarán entre el alcance pedido.

Deben respetarse todos los demás controles de servicio, pero su empleo no exige consideración especial alguna en el entorno distribuido.

17.4.5 *Ampliaciones (o extensiones)*

17.4.5.1 Si un ASG encuentra una operación-abstracta ampliada (extendida) en la misma fase resolución de nombre del procesamiento y determina que la operación abstracta debe concatenarse a uno o más ASG diferentes, deberá incluir sin modificarlas, en la operación abstracta concatenada, todas las ampliaciones presentes.

*Nota* - Una autoridad administrativa puede deteminar que es apropiado retornar un **ServiceError** con el problema **unwillingToPerform** si no desea propagar una ampliación.

17.4.5.2 Si un ASG encuetra una ampliación en una fase de ejecución del procesamiento, surgen dos casos posibles. Si la ampliación no es crítica, el ASG la ignorará. Si la ampliación es crítica el ASG retornará un **ServiceError** con el problema **unavailableCriticalExtension.**

Una ampliación crítica a una operación de objeto múltiple puede dar lugar a resultados y errores de servicio de esta variedad. Un ASG que fusiona esos resultados y errores descartará estos errores de servicio y empleará el componente **unavailableCriticalExtensión** del **PartialOutcomeQualifier** como se describe en el § 10.1.1 de la Recomendación X.511.

17.4.6 *Desreferenciación de alias*

La desreferenciación de alias es el proceso de creación de un nombre de nuevo objeto buscado, mediante la sustitución de la parte nombre distinguido de asiento de alias del nombre de objeto buscado original por el valor de atributo de nombre de objeto con alias a partir del asiento de alias. En la operación, el nombre de objeto no se ve afectado por la desreferenciación de alias.

17.5 *Autenticación de operaciones distribuidas*

Los usuarios de la guía junto con las autoridades administativas que proporcionan los servicios de guía pueden, si lo desean, exigir que se autentiquen las operaciones de guía. Para cualquier operación de guía particular, la naturaleza del proceso de autenticación dependerá de la política de seguridad en vigor.

Se dispone de dos conjuntos de procedimientos de autenticación que, aunados, permiten satisfacer una variedad de exigencias de autenticación. Uno de esos conjuntos está constituido por los procedimientos Vincular que facilitan la autenticación entre dos entidades de aplicación de guía para el establecimiento de una asociación. Los procedimientos Vincular permiten diversos intercambios de autenticación, desde un simple intercambio de identidades a una autenticación fuerte.

Además de la autenticación de la entidad par de una asociación, proporcionada por Vincular, se han definido procedimientos adicionales en la guía que permiten autenticar operaciones individuales. Están definidos dos conjuntos distintos de procedimientos de autenticación de guía. Uno de ellos facilita servicios de autenticación de originador que tratan de la autenticación, por un ASG, del iniciador de la petición de servicio original. El otro facilita servicios de autenticación de resultados que tratan de la autenticación, por un iniciador, de cualquier resultado que se le retorne.

Para la autenticación de originador se ha definido dos procedimientos. Uno de ellos se fundamenta en un simple intercambio de identidades y se le denomina autenticación basada en la identidad. El otro, en técnicas de firma digital y se llama autenticación basada en la firma. El primero de estos procedimientos es más bien rudimentario ya que el intercambio de identidad se basa en el intercambio de nombres distinguidos que se transmiten en claro.

Para autenticar resultados se ha definido un único procedimiento de autenticación de resultados basado en técnicas de firma digital. Debido al carácter generalmente complejo de la contrastación de resultados, no se ha definido para su autenticación ningún procedimiento que se base en la identidad.

Estos procedimientos no facilitan la autenticación de respuestas de error.

Debe considerarse que los servicios antes descritos aumentan los proporcionados por el servicio Vincular. Se supone que los procedimientos Vincular se han seguido de manera satisfactoria antes de proceder a la autenticación de operaciones de guía.

En el § 18.9 se especifican los procedimientos que debe seguir un ASG al proporcionar autenticación de originador y de resultados.

## 18 Comportamiento del ASG

### 18.1 *Introducción*

En correspondencia con cada operación invocada por un peticionario (un AUG o un ASG) el ASG actuante debe comportarse según procedimientos bien definidos, de tal modo que una respuesta adecuada se retorne de forma determinística. En este punto se especifica el commportamiento permitido, modelando un ASG desde el punto de vista de los procesos de ejecución de un determinado conjunto de procedimientos. Es importante tener en cuenta que un ASG debe únicamente ajustarse al comportamiento externo visible implicado por esos procedimientos y no a los propios procedimientos.

### 18.2 *Visión de conjunto del comportamiento de ASG*

El comportamiento global de la guía distribuida es la suma de los comportamientos de sus ASG cooperantes. Puede contemplarse cada uno de esos ASG como un proceso soportado internamente por un conjunto de procedimientos.

En la figura 6/X.518 se ilustra la visión interna del comportamiento de un ASG.

El despachador de operaciones es el principal procedimiento controlador de un ASG. Dirige cada operación a través de las tres fases descritas en el § 17.2.

Los procedimientos que facilitan la tarea del despachador de operaciones son: Resolución de nombre, Encontrar contexto de denominación, Resolución de nombre local, Evaluación de un solo objeto, Evaluación de múltiples objetos, y Fusión de resultados. Las relaciones entre estos procedimientos se muestran gráficamente en la figura 6/X.518.

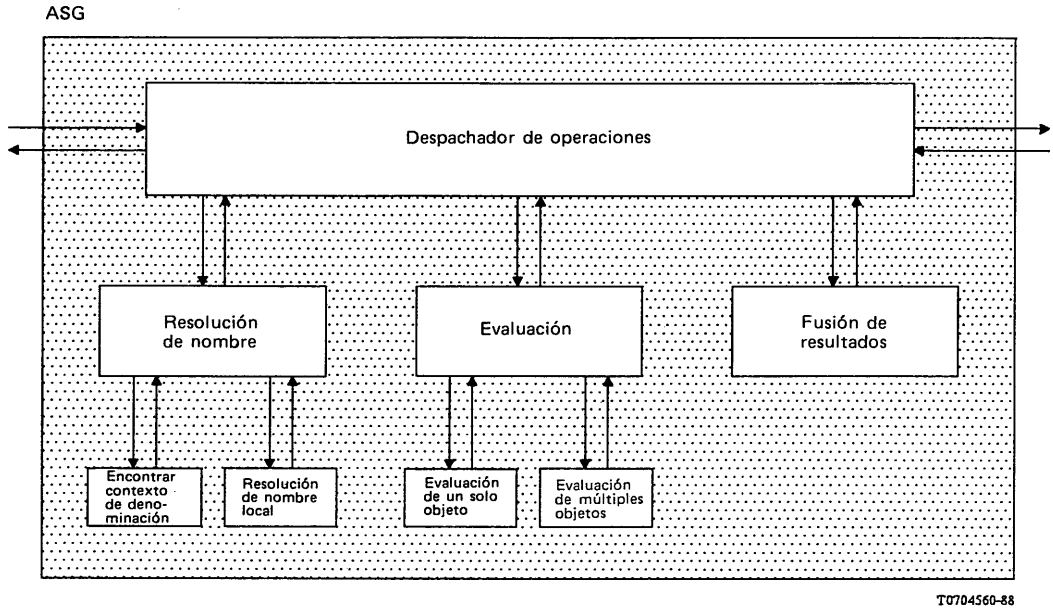



FIGURA 6/X.518

**Comportamiento de ASG - Vista interna**

18.2.1 *El despachador de operaciones*

Al recibir una operación, el despachador de operaciones comienza por validarla haciendo una comprobación relativa a bucles o errores de autenticación. Si no se encuentra nada, llama al procedimiento de resolución de nombre que retorna una indicación de encontrado o de error o una referencia. Las referencias se tratan por un referimiento o por una acción de concatenación o de difusión; las indicaciones de encontrado, llamando al procedimiento de evaluación que es el que realiza la operación deseada. Una vez retornados, los resultados internos y externos son cotejados por el procedimiento de fusión de resultados y, en ausencia de errores, enviados al AUG o ASG llamante.

18.2.2 *Resolución de nombre*

Resolución de nombre llama a encontrar contexto de denominación. Si el contexto retornado es local, se llama a resolución de nombre local; en otro caso, resolución de nombre retorna una referencia o un error y termina. Si resolución de nombre local encuentra un alias, éste es desreferenciado (de estar permitido) y resolución de nombre repite el análisis desde el principio. En otro caso, resolución de nombre local retorna una indicación de encontrado, un error o un referimiento que es devuelto al despachador de operaciones.

18.2.3 *Encontrar contexto de denominación*

Encontrar contexto de denominación trata de concordar el nombre contemplado con prefijos de contexto. Si ninguno concuerda, Encontrar contexto de denominación intenta identificar una referencia recíproca o superior. Si encuentra concordancia con un prefijo de contexto, retorna una referencia recíproca de relación descendente en el AIG o una indicación de que se encontró localmente un contexto de denominación adecuado y fija la "fase de resolución de nombre" en "en curso" ("proceeding").

18.2.4 *Resolución de nombre local*

El procedimiento de resolución de nombre local trata de concordar NDR del nombre contemplado, internamente, hasta que pueda retornar una indicación de encontrado. Si es incapaz de hacer concordar todos los NDR internamente, intenta identificar referencias subordinadas específicas primero, y luego no específicas, y las retorna a resolución de nombre. Si se encuentra un alias y la desreferenciación está permitida por los controles de servicios, se retorna una indicación de alias desreferenciado. De no estar permitida la desreferenciación se retornaría una indicación de encontrado, únicamente, si todos los NDR hubieran concordado en el momento en que se encontró el alias, sinó se devuelve un **nameError**.

18.2.5 *Evaluación*

El procedimiento de evaluación es el que finalmente realiza la operación de guía solicitada en el objeto buscado. Se invoca evaluación de un solo objeto o evaluación de múltiples objetos según cuál sea el tipo de operación.

18.2.6 *Evaluación de un solo objeto*

El procedimiento de evaluación de un solo objeto se invoca para **Read**, **Compare** o **AddEntry**, **RemoveEntry**, **ModifyEntry** y **ModifyRDN**. Es en este procedimiento en donde se extraen, verifican o cambian, efectivamente, los atributos.

18.2.7 *Evaluación de múltiples objetos*

Este procedimiento se invoca para las operaciones de **Search** y **List**, a fin de verificar filtros, extraer resultados y, si es preciso, despachar subpeticiones.

18.2.8 Fusión de resultados

El procedimiento de fusión de resultados coteja resultados o errores recibidos de otros ASG los resultados extraídos localmente.

18.3 *Operaciones específicas*

Las operaciones pueden clasificarse en tres categorías (la operación y su contrapartida concatenada están siempre en la misma categoría).

a) Operaciones de un solo objeto: **Read, Compare, AddEntry, ModifyEntry, Modify RDN, RemoveEntry.**

b) Operaciones de múltiples objetos: **List, Search.**

c) Operación de abandono, es decir, **Abandon.**

El tratamiento de estas tres categorías se describe en los § 18.3.1 a 18.3.3 respectivamente. Puesto que hay una gran similitud entre el comportamiento de un ASG al efectuar una operación de puerto-servicio y al efectuar su operación de contrapartida concatenada de un puerto-servicio concatenado, se hace una sola descripción que se aplica a ambas operaciones señalándose, en su caso, las excepciones a esta regla.

18.3.1 *Operaciones de un solo objeto*

Operaciones de un solo objeto son las que afectan a un solo asiento y que, por lo tanto, se pueden llevar totalmente a cabo en el ASG que contiene el asiento sobre el que se realiza la operación. Tales operaciones se describen habitualmente por la siguiente secuencia de sucesos:

1) Activar el despachador de operaciones.

2) Seguir el procedimiento de resolución de nombre para localizar el objeto cuyo nombre se ha especificado como argumento de la operación.

3) Seguir el procedimiento de evaluación de un solo objeto.

4) Los controles de servicios tales como el de límite de tiempo, deberán verificarse en el transcurso de la operación para imponer las limitaciones especificadas por el usuario.

5) Retornar los resultados al AUG o ASG que transmitió la petición.

18.3.2 *Operaciones de múltiples objetos*

Operaciones de múltiples objetos son las que afectan a varios asientos que pueden estar co-ubicados o no en el mismo ASG. Tales operaciones pueden así entrañar un esfuerzo cooperativo por parte de varios ASG para localizar todos los asientos afectados por la operación pedida y actuar en los mismos. El comportamiento habitual de tales operaciones se puede resumir en las siguientes actuaciones:

1) Activar el despachador de operaciones.

2) Seguir los procedimientos de resolución de nombre para localizar el objeto cuyo nombre se especificó como argumento de la operación.

3) Una vez que se ha localizado el objeto buscado de la operación, seguir los procedimientos de evaluación de objetos-múltiples.

4) Si se ha producido una descomposición de petición en alguno de los procedimientos de evaluación de objetos-múltiples y, subsiguientemente, se han concatenado/difundido subpeticiones, el despachador de operaciones retiene los resultados locales actuales, aguarda las respuestas concatenadas y activa la fusión de resultados.

5) Los controles de servicios, tales como el límite de tiempo y límite de tamaño, deberán ser verificados en el transcurso de la operación para mantenerse dentro de las limitaciones especificadas en el argumento común.

6) Retornar los resultados o errores al AUG o el ASG que transfirió la petición.

18.3.3 *Operación de abandono*

Al recibir una operación de abandono, un ASG determina si puede o no abandonar la operación especificada y, en caso positivo, la abandona y devuelve un resultado (la operación que fue abandonada retorna un error **Abandoned**). Si no puede abandonar la operación especificada, retorna un error **AbandonFailed**.

A continuación se indica el procedimiento específico de la operación de **Abandon**.

1) Localizar la operación cuyo identificador de invocar se especifica como argumento de la operación de **Abandon**.

2) Facultativamente, componer petición (peticiones), con el adecuado identificador-invocar, para abandonar todas las operaciones pendientes, concatenadas o difundidas a otros ASG.

3) Facultativamente la operación de abandonar se realiza localmente, tal como se define en la Recomendación X.511.

4) Retornar resultado o error al AUG o el ASG que transmitió la petición.

18.4 *Despachador de operaciones*

18.4.1 *Introducción*

El despachador de operaciones utiliza el procedimiento de resolución de nombre descrito en el § 18.6 de esta Recomendación más todas las interacciones (es decir, de ASG a ASG o de AUG a ASG) necesarias para localizar asientos objetivo en un entorno de guía distribuida. La figura 7/X.518 muestra, en forma de diagrama detallado, una descripción del despachador de operaciones. Su algoritmo se resume más adelante.

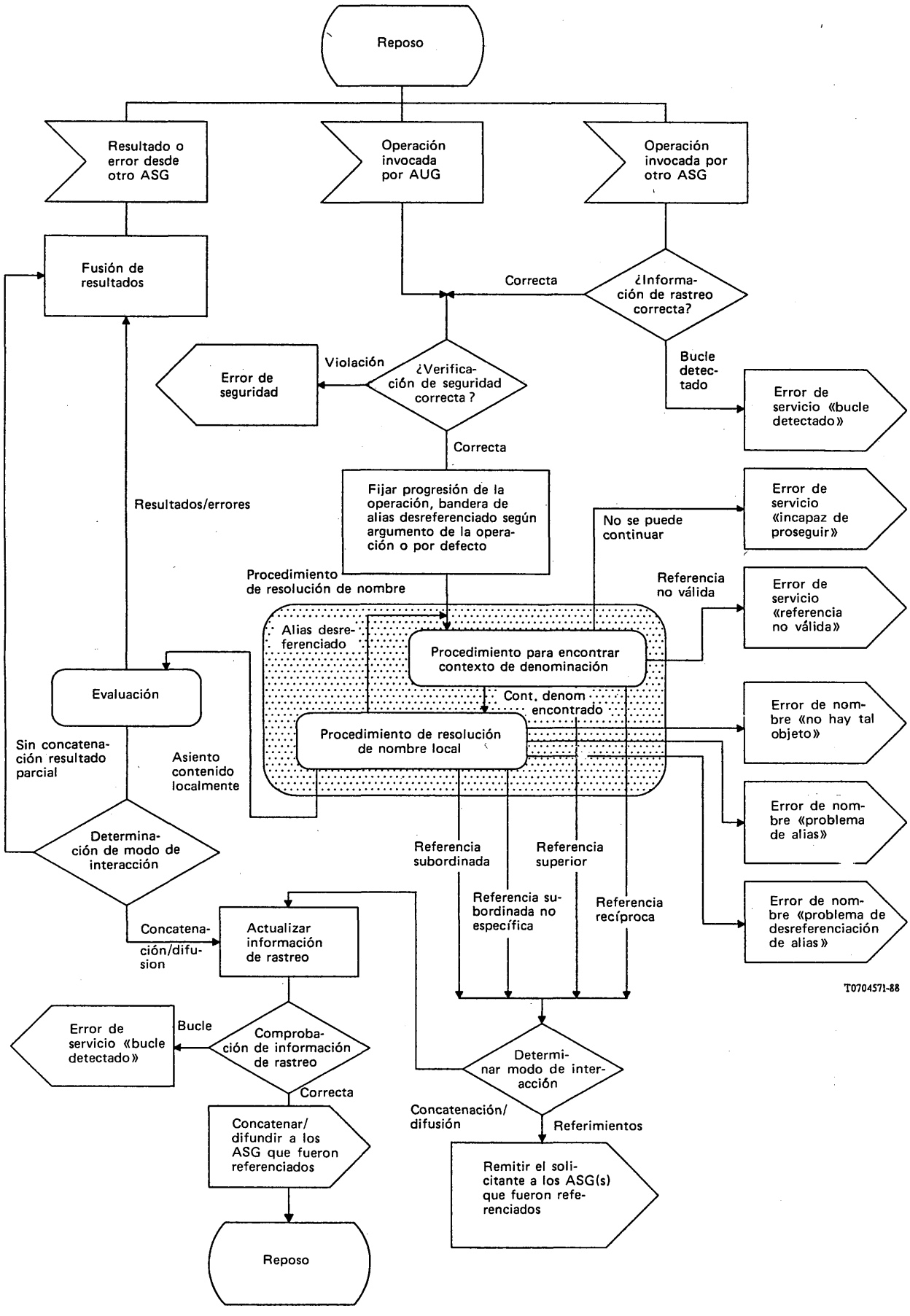



FIGURA 7/X.518

**Despachador de operaciones**

18.4.2 *Acciones implícitas*

18.4.2.1 *Seguridad*

Hay que tener en cuenta que, si bien la verificación de firmas no se incluye de manera explícita en este algoritmo, se trata de una acción que es siempre el primer paso cuando una operación, un resultado o un error firmados llegan al ASG.

*Nota* - Esto no incluye firmas insertadas.

Si la firma no es válida o no figura en un caso en que debería estar presente, se retorna un **securityError**. Todo el tratamiento de la operación se concluye y el despachador de operaciones va a su estado de reposo.

La firma de un resultado de operación, de ser necesaria es, de forma similar, un último paso implícito antes de despacharla.

18.4.2.2 *Controles de servicios (Service Controls)*

Aunque los **ServiceControls** no se mencionan de manera explícita, sí en cambio se respetan. Por ejemplo, se considera obligatoria la verificación del **timeLimit** de una operación que llega y la del **SizeLimit** antes de enviar un resultado. Estos controles se examinan en el § 17.4.4.

18.4.2.3 *Información de rastreo (Trace Information)*

Siempre se actualiza la **TraceInformation** con el estado con que llegó al ASG antes de incluirla en los **ChainingArguments**. En el texto que sigue, no se indica esto de forma explícita.

18.4.3 *Argumentos*

Argumentos de concatenación para la operación en cuestión.

18.4.4 *Resultados*

Resultados de concatenación para la operación en cuestión.

18.4.5 *Errores*

Cualquier error definido en esta Recomendación.

18.4.6 *Algoritmo*

1) Recibir operación.

   Si la operación procede de otro ASG, contendrá los argumentos de concatenación, incluidos los de **operationProgress**, **aliasDereferenced**, **aliasedRDNs**, **targetObject Name** y **TraceInformation**, así como los parámetros contenidos en la operación original.

   Si procede de un AUG no contendrá la indicación de **aliasDereferenced**: adóptese el valor **FALSE**. En el argumento tampoco figura ninguna **TraceInformation**, por lo que no es preciso verificar la existencia de bucles. Establecer como nombre de **targetObject Name** el del objeto buscado para la operación (véase el § 17.2). Otros argumentos de concatenación se fijan de acuerdo con los parámetros de la operación DAP. El **originator** se fija al nombre del usuario.

2) Si la operación vino de un ASG, se verifica la información de rastreo para evitar bucles (activar Detección de bucles). Si se detecta un bucle, se retorna **ServiceError** con un problema de **loopDetected** y se termina el tratamiento.

3) Se efectúan comprobaciones de seguridad de la operación (un origen en un AUG o en un ASG). Si se detecta una violación, se retorna un **error de seguridad**. En otro caso, se fija **operationProgress** y **aliasDereferenced** de acuerdo con el argumento de la operación, o por defecto.

4) Ejecutar el procedimiento de resolución de nombre.

   El procedimiento resolución de nombre retornará una indicación de encontrado, una referencia distante o una indicación de error.

5) Se puede provocar alguno de los siguientes errores:

   **ServiceError (UnableToProceed)** si un ASG detecta que se le transfirió una operación relacionada con una información que él no contiene.

**ServiceError (InvalidReference)** - si un ASG determina que se utilizó una referencia de conocimiento no válida.

**NameError (noSuchObject)** - si se determina que el nombre contemplado especificado en la petición de operación no es válido.

**NameError (aliasProblem)** - si se ha desreferenciado un alias que no nombra ningun objeto.

**NameError (aliasDereferencingProblem)** - si se encontró un alias en una situación no autorizada.

Al recibirse cualquiera de estos errores, el despachador de operación termina y se retorna un error al ASG o al AUG que dio origen a la operación distribuida.

6) Si se retorna encontrado (Found) , se activa el procedimiento de evaluación.

7) Si se retorna una referencia distante (de un procedimiento de resolución de nombre o de una evaluación) podrá ser de alguno de los siguientes tipos: referencia recíproca, referencia subordinada, referencia superior o referencia subordinada no específica.

   El retorno de alguna de esas referencias significa que los procedimientos de resolución de nombre o evaluación no se pueden completar en el ASG de que se trate, por lo que deberá participar en los procedimientos el ASG identificado en la referencia.

   El despachador de operaciones determina a continuación el modo de interacción: por referimiento o por concatenación.

8) Si se selecciona el modo de interacción por referimiento, dependiendo del **scopeOfReferral**, la información contenida en la referencia retornada se remite al AUG o al ASG de origen como un referimiento, o se devuelve **OutOfScope ServiceError**. El tratamiento de la operación termina entonces.

*Nota* - Si **returnCrossRefs** es verdadero y la referencia no es una referencia subordinada no específica ni una referencia superior y, además, la autoridad administrativa está dispuesta a proporcionar conocimiento, podrá fijarse el prefijo de contexto en el referimiento.

9) Si se selecciona el modo de interacción por concatenación, la operación se transfiere al ASG especificado en la referencia. En caso de referencia subordinada no específica, la operación ha de transferirse a cada uno de los ASG cuyo nombre se alcanzó como parte de una referencia subordinada no específica. Esa transferencia puede realizarse por difusión o por concatenación sucesiva de la operación.

10) Aplicar la estrategia de prevención de bucles (**Loop Avoidance**) con cada operación que se va a enviar. Si la prevención no resulta aplicable o no se detecta bucle, asignar valores a los argumentos de concatenación incluyendo una versión actualizada de **traceInformation** y enviar las operaciones.

    Si no se enviaran operaciones (pòr problemas de formación de bucle), retornar un **serviceError** (con problema de **loopDetected**) y concluir el tratamiento de la operación.

*Nota* - Si en este paso la operación descompuesta es abortada para evitar bucles, será un asunto local el devolver un resultado parcial o el abortar la operación completa y retornar un error. Si se elige esto último, retornar **ServiceError** con el problema **LoopDetected** y concluir el tratamiento.

11) Esperar las respuestas y realizar luego el procedimiento de fusión de resultados.

## 18.5 *Formación de bucles*

En el contexto de una operación de guía particular se produce un bucle si, en cualquier momento, la operación vuelve a un estado anterior (según se define en el § 17.4.2). Esto no significa que una operación no pueda ser procesada múltiples veces por un determinado ASG pero si significa que el ASG no procesará la misma operación en el mismo estado varias veces.

Para el tratamiento de la formación de bucles se utiliza el argumento de información de rastreo definido en el § 12.6. Se han elaborado dos estrategias con respecto a los bucles: la de detección y la de prevención de bucles descritas en los § 18.5.1 y 18.5.2 respectivamente.

18.5.1 *Detección de bucles*

La detección de bucles requiere que al recibir un ASG una operación entrante determine si el estado que en ese momento tiene la operación figura en la sucesión de estados anteriores registrados en su argumento traceInformation. Si efectivamente figura, la operación está en bucle, retornándose un **serviceError** (con problema de **loopDetected**). En caso contrario, el ASG continúa tratando la operación según los procedimientos especificados en el § 18.4.

18.5.2 *Prevención de bucles*

La prevención de bucles requiere que un ASG determine inmediatamente antes de transferir una operación a otro ASG (como parte de una concatenación, de una difusión, o de un procedimiento de descomposición de petición), si el consiguiente estado de la operación (si se conoce) figura en la serie de estados anteriores registrada en el argumento **traceInformation** de la operación entrante original. El estado consiguiente es el valor de **TraceItem** que el ASG receptor añade a **TraceInformation.**

En el caso de que la operación entrante original estuviera en un puerto-servicio (y no en un puerto-servicio concatenado) no habrá información de rastreo y el procedimiento de prevención de bucle será intrascendente.

Si se conoce el estado consiguiente de la operación, y aparece en la **traceInformation,** la invocación de la operación provocaría un bucle. En este caso, la respuesta adecuada a la operación original es un **serviceError** (con problema de **loopDetected).**

18.6 *Procedimiento de resolución de nombre*

En este punto se describe detalladamente el procedimiento de resolución de nombre, sus parámetros de entrada y salida y sus posibles condiciones de error. La figura 7/X.518 muestra el procedimiento global en forma de diagrama. El procedimiento de resolución de nombre llama a dos procedimientos componentes:

1) Encontrar contexto de denominación (figura 8/X.518).

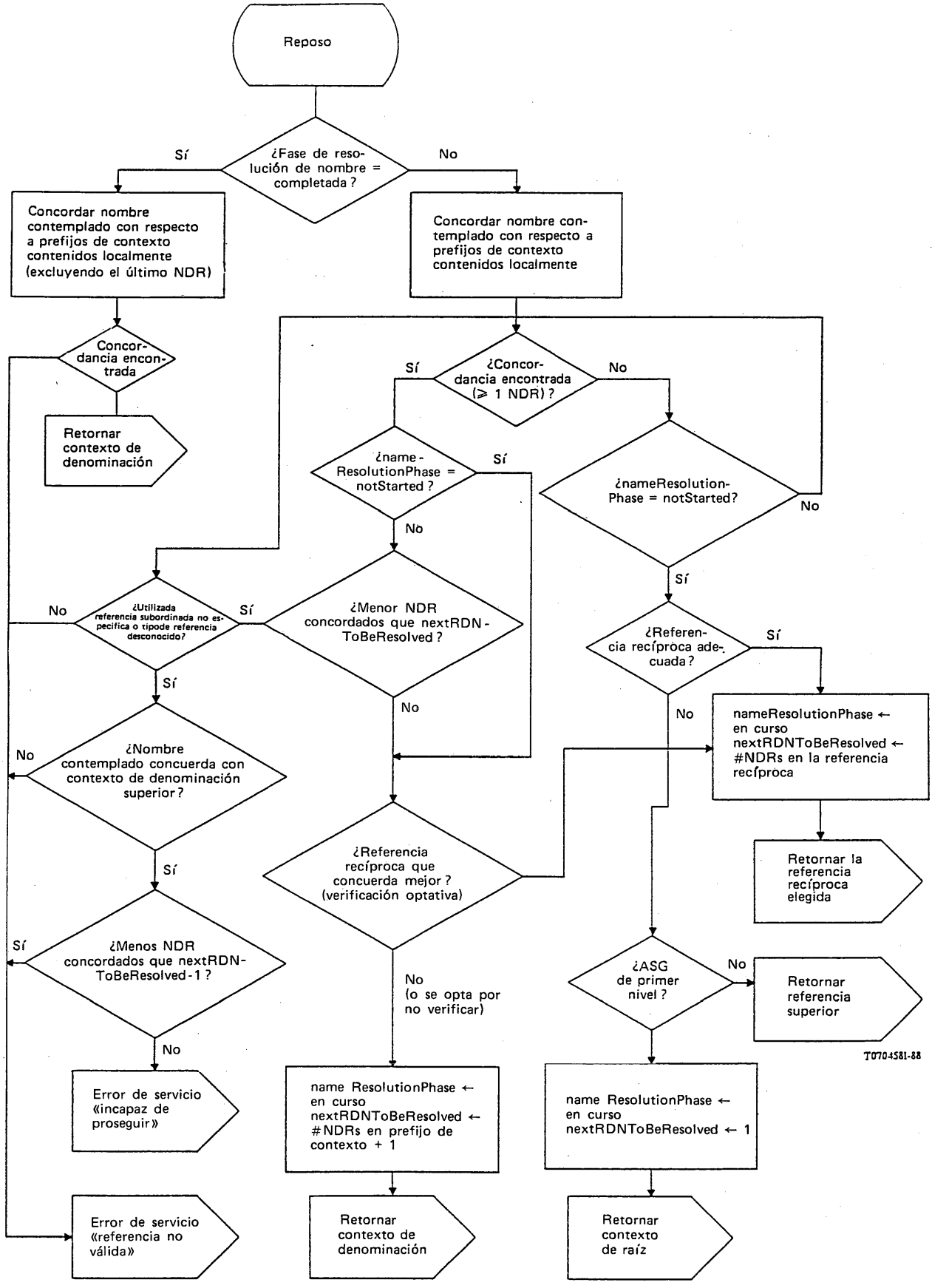



FIGURA 8/X.518

**Contexto de encontrar denominación**

2) Resolución de nombre local (Figura 9/X.518).

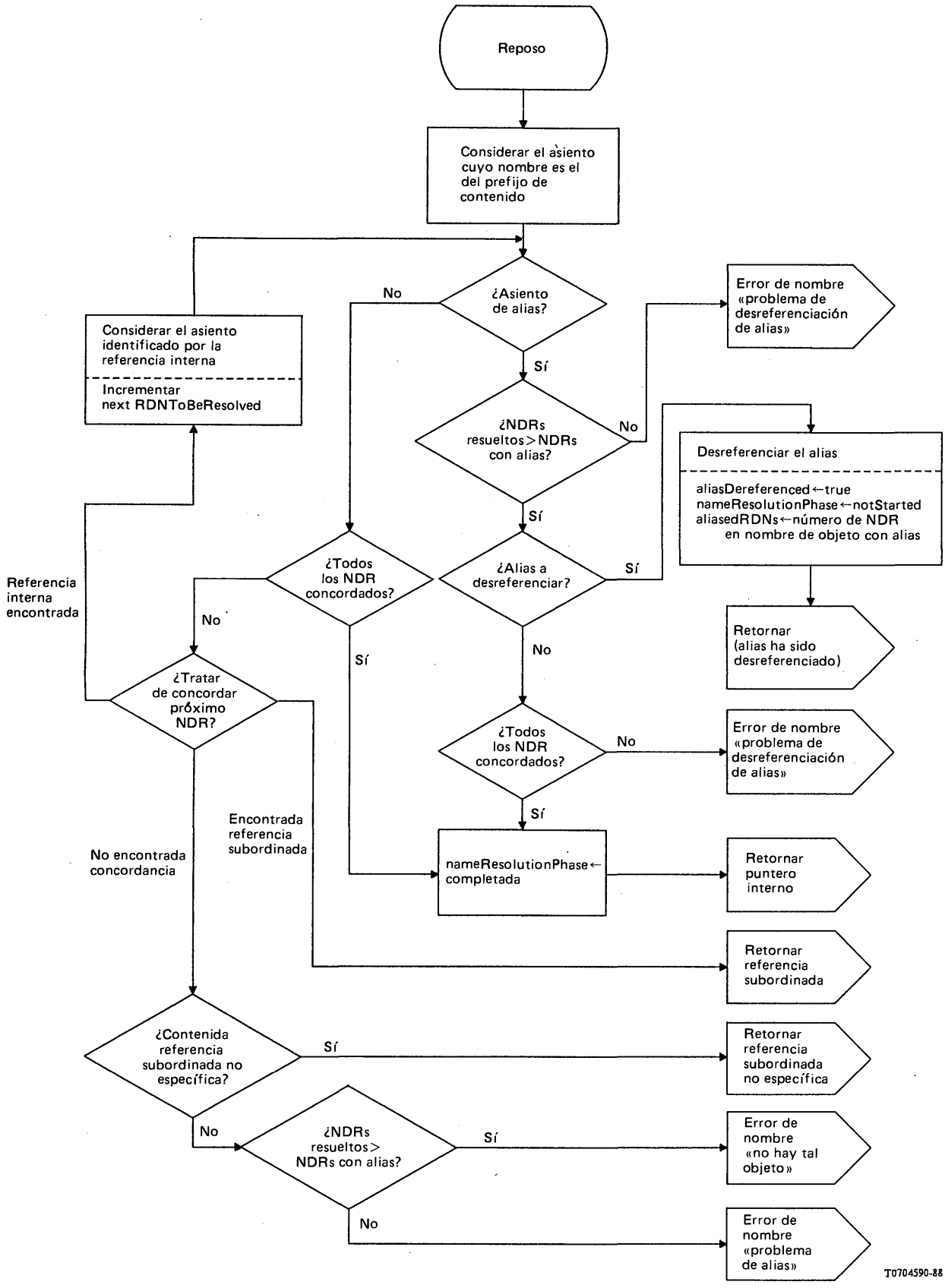



FIGURA 9/X.518

**Resolución de nombre local**

El procedimiento de resolución de nombre devuelve al despachador de operaciones los resultados de esos procedimientos componentes excepto en dos casos. El primero se da cuando el procedimiento de encontrar contexto de denominación identifica un contexto adecuado que requiere un examen ulterior y retorna el contexto de denominación local. El segundo se da cuando el procedimiento resolución de nombre local indica que ha desreferenciado un alias. En el primer caso, el procedimiento resolución de nombre llama al procedimiento resolución de nombre local. En el segundo caso, el procedimiento resolución de nombre se reactiva con el nuevo nombre de objeto buscado.

18.6.1 *Argumentos*

El procedimiento utiliza los siguientes argumentos:

- el nombre de objeto buscado (el nombre contemplado);
- avance de la operación;
- el valor del control de servicio **dontDereferenceAliases**;
- el valor del parámetro **aliasedRDNs**;
- el valor del parámetro **aliasDereferenced**.

18.6.2 *Resultados*

Hay dos posibles resultados satisfactorios.

Con el primero se retorna:

- una referencia;
- avance de la operación (convenientemente actualizado);
- indicación de **aliasDereferenced** y, facultativamente, **aliasedRDNs**.

Con el segundo se retorna:

- una indicación de que se ha encontrado el contexto de denominación (junto con el puntero local que señala hacia el asiento);
- avance de la operación (convenientemente actualizado);
- indicación de **aliasDereferenced** y, facultativamente, **aliasedRDNs**.

18.6.3 Errores

Puede retornarse uno de los siguientes errores:

- **ServiceError (unableToProceed)**;
- **ServiceError (invalidReference)**;
- **NameError (aliasProblem, noSuchObject,** o **alias Dereferencing Problem)**.

18.6.4 *Procedimiento*

1) Activar el procedimiento de encontrar contexto de denominación.
2) Aguardar respuesta del procedimiento de encontrar contexto de denominación.
3) Recibir resultados o error retornados, a saber: contexto de denominación local encontrado, referencia distante, error de incapaz de proseguir o error de nombre o referencia no válida.
4) Realizar funciones basadas en los resultados o el error retornados.
   a) Si se ha encontrado el contexto de denominación local, activar el procedimiento de resolución de nombre local. Este procedimiento puede retornar una indicación de referencia interna encontrada, de referencia distante, de desreferencia de alias o un error de nombre. Cada una de estas respuestas da lugar a que concluya la resolución de nombre con el resultado notificado, pero si se ha desreferenciado un alias, el procedimiento vuelve a comenzar en el paso 1).
   b) Cualquier otro resultado se pasa al despachador de operaciones.

18.6.5 *Procedimiento de encontrar contexto de denominación*

18.6.5.1 *Introducción*

En la figura 8/X.518 se muestra este procedimiento en forma de diagrama. Lo que viene a continuación es una descripción textual del procedimiento. En ésta se supone que el valor corriente de avance de la operación se devuelve siempre después de terminado el procedimiento.

18.6.5.2 *Argumentos*

El procedimiento utiliza los siguientes argumentos:

- el nombre de objeto buscado (el nombre contemplado);
- avance de la operación.

18.6.5.3 *Resultados*

Hay dos posibles resultados satisfactorios.

Con el primero se retorna:

- una referencia;
- avance de la operación (convenientemente actualizado).

Con el segundo se retorna:

- una indicación de que se encontró localmente un contexto de denominación adecuado;
- avance de la operación (convenientemente actualizado).

18.6.5.4 *Errores*

Se puede retornar uno de los siguientes errores:

- **ServiceError (unableToProceed);**
- **ServiceError (invalidReference).**

18.6.5.5 *Procedimiento*

1) Si la **nameResolutionPhase** se pone a **completed** en un asiento, intentar la concordancia del nombre contemplado con todos los prefijos de contexto de los contextos de denominación superiores de todos los contextos de denominación contenidos localmente. Si se encuentra una concordancia, retornar todos los contextos de denominación contenidos localmente. Si no se encuentra concordancia, retornar un **serviceError** de **invalidReference.**

2) Si la **nameResolutionPhase** no está puesta a **completed,** tratar de hallar concordancia entre los prefijos de contexto y una sucesión de uno o más NDR de la porción inicial del nombre contemplado. Para encontrar una concordancia, todos los NDR de un prefijo de contexto deben estar concordados. Los prefijos de contexto utilizados son los de contextos de denominación para los que el ASG tiene autoridad administrativa. Si se producen múltiples concordancias, se elige el prefijo que concuerda con mayor número de NDR.

   Si se encuentra una concordancia, ejecutar 3).

   Si no se encuentra ninguna concordancia, ejecutar 5).

3) Si la **nameResolutionPhase** está **notStarted,** ejecutar 4). Si el número de NDR es la porción inicial del nombre contemplado, concordado como se ha descrito en 2), es mayor o igual que el componente **nextRDNToBeResolved** de **operationProgress,** ejecutar 4), y en otro caso ejecutar 9).

4) El **nextRDNToBeResolved** se fija al número de NDR concordados más 1, y la **nameResolutionPhase,** a **proceeding.** Se retorna el contexto y se da por terminado el procedimiento.

   Como una mejora del funcionamiento, el ASG puede, facultativamente, concordar el nombre contemplado con respecto a referencias recíprocas que él mismo contiene. Si son concordados más NDR con respecto a la referencia recíproca que con respecto a los prefijos de contexto contenidos localmente, ejecutar el paso 7).

*Nota* - En el caso de que se produzca este resultado, el procedimiento de resolución de nombre llamará al de resolución de nombre local.

5) Si no se encontró ninguna concordancia, se verifica el valor de **nameResolutionPhase**. Si la **nameResolutionPhase** es **notStarted**, ejecutar 6).

   Si el valor de **nameResolutionPhase** es **proceeding** o **completed**, ejecutar 9).

6) Utilizando prefijos de contexto de referencia recíproca, tratar de hallar su concordancia con una secuencia de uno o más NDR de la porción inicial del nombre contemplado. Si se producen múltiples concordancias, se elige el prefijo que concuerda con el mayor número de NDR.

7) Si se encontró concordancia para una referencia recíproca, fijar el **nextRDNToBeResolved** a número de NDR de la referencia recíproca elegida. Se retorna la referencia recíproca y se da por terminado el procedimiento.

8) Si no se encontró concordancia con una referencia recíproca, determinar si el ASG es de primer nivel. Si no lo es, tendrá una referencia superior. Retornar ésta y terminar el procedimiento.

   Si el ASG es un ASG de primer nivel, fijar **nextRDNToBeResolved** a uno, y **nameResolutionPhase** a **proceeding**. Retornar el contexto de denominación de raíz y terminar el procedimiento.

9) Verificar el valor del componente **referenceType** del **chainingArgument**. Si se usó una referencia subordinada no específica o la petición venía de un AUG, ejecutar 10); si no, retornar el **serviceError** con el problema **invalidReference** y terminar el procedimiento.

10) Comparar la parte inicial del nombre contemplado con los prefijos de contexto (menos su último NDR) de los contextos de denominación contenidos localmente. Esta es, en efecto, una comparación con algunos de los contextos de denominación del superior inmediato a este ASG.

    Si no hay concordancia, devolver **serviceError** con **invalidReference** y terminar el procedimiento.

    Si se encuentra concordancia, y si el número de NDR concordados es menor que en **nextRDNToBeResolved**-1, retornar **serviceError** con **invalidReference**; en otro caso, retornar **serviceError** con **unableToProceed**. Terminar el procedimiento.

### 18.6.6 *Resolución de nombre local*

#### 18.6.6.1 *Introducción*

El procedimiento de resolución de nombre local concuerda los NDR del nombre contemplado con referencias de conocimiento internas. Retorna indicaciones de encontrado, referencia distante, alias desreferenciado o error.

En la figura 9/X.518 se muestra este procedimiento en forma de diagrama. A continuación se hace una descripción textual del procedimiento.

#### 18.6.6.2 *Argumento*

El procedimiento utiliza los siguientes argumentos:

- referencia interna a contexto de denominación (con puntero que señala el asiento cuyo nombre coincide con el prefijo de contexto);
- el nombre de objeto buscado (el nombre contemplado);
- progresión de operación;
- el valor de control de servicio **dontDereferenceAliases**;
- el valor del parámetro **aliasedRDNs**;
- el valor del parámetro **aliasDereferenced**.

#### 18.6.6.3 *Resultados*

Hay tres casos de resultado con éxito.

En el primero se retorna:

- una referencia;

- progresión de operación (convenientemente actualizado).

En el segundo se retorna:

- una indicación de que el asiento se encontró localmente;
- progresión de operación (convenientemente actualizado).

En el tercero se retorna:

- una indicación de que se desreferenció un alias;
- progresión de operación convenientemente actualizado (repuesto a "no comenzado").

18.6.6.4 *Errores*

Puede retornarse uno de los siguientes errores:

- error de nombre.

18.6.6.5 *Procedimiento*

El contexto de denominación retornado por FindNamingContext apuntará al asiento de la raíz del subárbol. En el caso del contexto de raíz, el asiento sólo es un asiento vacío.

1) Si la referencia interna es para un asiento de alias, ejecutar el paso 7), y si no el paso 2).

2) Si han sido concordados todos los NDR en el nombre contemplado, el asiento deseado ha sido encontrado. Fijar **nameResolutionPhase** a **completed.** Se retorna un puntero interno y se da por terminado el procedimiento.

   En otro caso, deberá ejecutar el paso 3).

   *Nota* - Podría alcanzarse la concordancia sólo con el prefijo de contexto o con ese prefijo más sucesivos NDR contenidos en referencias internas en el árbol de conocimiento.

3) Si en el árbol de conocimiento se encuentra un asiento de referencia interna, subordinado al asiento corriente, que concuerda con el siguiente NDR del nombre contemplado, incrementar **nextRDNToBeResolved**, establecer el asiento corriente a asiento subordinado, y volver a ejecutar el paso 1) de este procedimiento.

4) Si el asiento corriente tiene una referencia subordinada cuyo NDR concuerda con el siguiente del nombre contemplado, retornarla y terminar el procedimiento.

5) Si hay algunas referencias subordinadas no específicas, subordinadas al asiento corriente en el árbol de conocimiento, retornarlas como referencias y terminar el procedimiento.

6) Si no se encuentra ninguna referencia interna, referencia subordinada, o referencia subordinada no específica, verificar cuántos NDR del nombre contemplado han sido concordados. Si han sido concordados más NDR que el componente **aliasedRDNs** de **chainingArgument**, devolver **NameError** con **noSuchObject**. Si han sido concordados menos NDR, devolver **NameError** con **aliasProblem**.

7) Si el número de NDR del nombre contemplado que han sido concordados es menor o igual que el componente **aliasedRDNs** de **ChainingArgument** (si existe), entonces el anterior alias que fue desreferenciado (en su caso) apunta a otro alias. Si esto es así, retornar **NameError** con **aliasDereferencingProblem**.

8) Si falta el componente **aliasedRDNs**, o si el número de NDR concordados es mayor que el componente **aliasedRDNs** del **ChainingArgument**, verificar el control de servicio **dontDereferenceAlias**. Si se puede desreferenciar alias, ejecutar el paso 9). Si no, ejecutar el paso 10).

9) Desreferenciar el alias. Fijar **nameResolutionPhase** de **OperationProgress** a **no comenzada**. Fijar el componente **aliasDereference** de **ChainingArgument** a **TRUE** y **aliasedRDNs** al número de NDR en el atributo **aliasedObjectName** del asiento con alias. Fijar targetObject al nuevo nombre. Terminar el procedimiento. (Se volverá a comenzar el proceso de resolución de nombre.)

10) Si todos los NDR en el nombre contemplado han sido concordados, ejecutar el paso 2). Si no, retornar **NameError** con **aliasDereferencingProblem**.

18.7 *Procedimientos de evaluación de objeto*

Hay dos categorías de procedimientos de evaluación de objeto:

a) los procedimientos de evaluación de un solo objeto;

b) los procedimientos de evaluación de múltiples objetos.

La figura 10/X.518 muestra los procedimientos de evaluación de objeto.

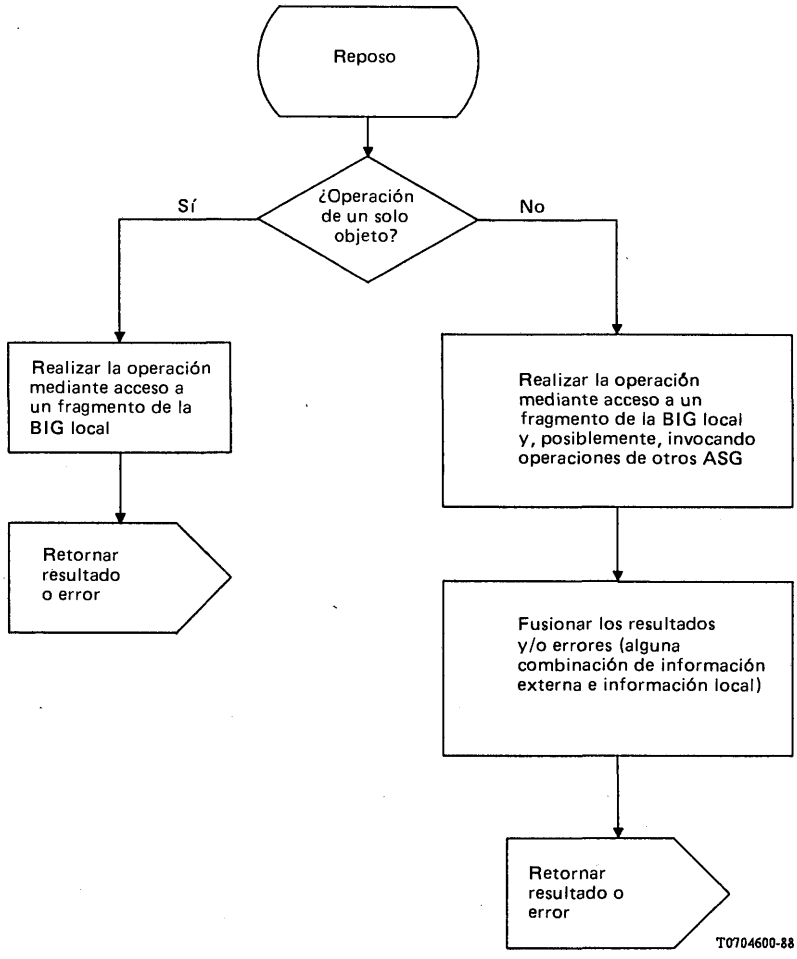



FIGURA 10/X.518

**Evaluación y fusión de resultados**

18.7.1 *Procedimiento de evaluación de un solo objeto*

Los procedimientos de evaluaión de un solo objeto, que son propios de las operaciones en las que se produce el acceso a un solo objeto, se realizan directamente, retornándose el resultado o el error al invocador.

Estas operaciones son las de **Read, Compare, AddEntry, RemoveEntry, ModifyEntry** y **ModifyRDN**, y sus contrapartidas concatenadas. La acción requerida para un asiento es la descrita en el punto apropiado de la Recomendación X.511.

Las operaciones **AddEntry, RemoveEntry**, y **ModifyRDN** afectan al conocimiento. Si el superior inmediato del asiento está en un ASG diferente, se mantendrán las referencias de conocimiento externas correctas. La forma de conseguirlo queda fuera del alcance de esta Recomendación.

La manera de elegir el ASG para que contenga el asiento creado por **AddEntry** queda fuera del alcance de esta Recomendación.

Si el superior inmediato de un asiento que haya que crear por **AddEntry** o modificar por **ModifyRDN** tiene referencias subordinadas no específicas, deberán seguirse procedimientos, que caen fuera del alcance de esta Recomendación, para asegurar que no habrá dos entradas con el mismo nombre distinguido.

Las peticiones que no puedan satisfacerse en estas condiciones fracasarán con un **UpdateError** con **affectsMultipleDSAs.**

18.7.2 *Procedimientos de evaluación de múltiples objetos*

Los procedimientos de evaluación de múltiples objetos, comunes a las operaciones en las que se produce acceso a múltiples objetos, se especifican en los párrafos siguientes.

Las referidas operaciones son las de **List** y **Search** y sus contrapartidas concatenadas.

18.7.2.1 *Listar*

En este punto se especifica el procedimiento de evaluación propio de **List** y **ChainedList.** (En lo que sigue, el término "listar" se refiere a ambas operaciones.)

18.7.2.1.1 *Procedimiento de listar (I)*

El procedimiento de listar (I) se aplica cuando la petición de listar tiene el componente **nameResolutionPhase** de **OperationProgress** fijado en **notstarted** o **proceeding** y cuando el ASG, tras efectuar la resolución de nombre, encuentra que contiene el objeto base.

El objeto base se designará por "e".

1) Obtener cada uno de los subordinados inmediatos de e contenidos localmente para formar un conjunto local de resultados. Establecer **aliasEntry** y **fromEntry** en **ListResult** en la forma adecuada.
2) Tomar el conjunto de referencias subordinadas no específicas y referencias subordinadas a los ASG que contienen subordinados inmediatos de "e".
3) Transferir la subpetición con el objeto de base = e y **OperationProgress** puesto a **"completada"** al despachador de operaciones, el cual la reenviará seguidamente a cada ASG que contienen subordinados inmediatos de e.

*Nota* - Si el ASG contiene referencias subordinadas con indicación de si el objeto subordinado tiene o no alias, y el **dontUseCopy** es **FALSE**, este paso puede entonces omitirse para esos asientos. La información sobre los subordinados está disponible directamente.

18.7.2.1.2 *Procedimiento de listar (II)*

El procedimiento de listar (II) se aplica a una petición de listar con el componente **nameResolutionPhase** de **OperationProgress** fijado a **completed.**

El objeto base se designará por "e".

1) Obtener cada uno de los subordinados inmediatos de e para formar un conjunto local de resultados. Fijar **aliasEntry** y **fromEntry** o **ListResult** de la manera adecuada.
2) Pasar los resultados al despachador de operaciones que los reenviará al AUG o ASG solicitante.

18.7.2.2 *Buscar*

En este párrafo se especifica el procedimiento de evaluación propio de Buscar y Concatenado Buscar. (En lo que sigue, el término "buscar" se refiere a ambas operaciones.)

Obsérvese que existen dos circunstancias que exigen dos procedimientos distintos. El primer procedimiento (§ 18.7.2.2.1) se aplica cuando el ASG que ejecuta la búsqueda contiene el **targetObject** en un asiento local. El segundo procedimiento (§ 18.7.2.2.2) se aplica cuando el ASG que ejecuta la búsqueda no contiene el **targetObject,** sino solamente subordinados del **targetObject.**

18.7.2.2.1 *Procedimiento de buscar (I)*

El procedimiento de buscar (I) se aplica a una petición de buscar con el componente **nameResolutionPhase** de **OperationProgress** fijado a **notStarted** o **proceeding** y cuando el ASG, tras efectuar la resolución de nombre, ve que él contiene el objeto deseado.

El objeto base se designará por "e".

1) Si el argumento del **subset** es **baseObject** o **wholeSubtree**, aplicar el argumento del filtro especificado en la búsqueda para el asiento e, para formar un conjunto de resultados locales. Retornar los resultados para fusión de resultados. Si el argumento del **subset** es **baseObject**, terminar el procedimiento; si no, pasar a 2).

2) Si el argumento del **subset** es **oneLevel** o **wholeSubtree**, formar un conjunto E a partir de los subordinados inmediatos de e mantenidos localmente, salvo que:

   Si los alias no son desreferenciados, es decir, el parámetro **searchAlias** es **TRUE**, todo asiento con alias será tratado de conformidad con el § 5) más abajo y no contribuirá a estos resultados.

   Aplicar los argumentos de filtro a E para que de un subconjunto filtrado E' de E; devolver dicho conjunto E' de resultados locales para fusión de resultados.

3) Otros subordinados de e pueden residir en otros ASG, en cuyo caso serán desreferenciados como referencias subordinadas o como referencias subordinadas no específicas. Para cada ASG que sea desreferenciado de esta forma, preparar una nueva búsqueda con targetObject = e, y con **nameResolutionPhase** de **OperationProgress** fijado a **completada**. Retornar cada subpetición de búsqueda al despachador de operaciones para que éste las reenvíe. Si se retorna un error con respecto a una subconsulta, ésta se ignora como si no se hubiese hecho.

4) Si el argumento de **subconjunto** es **oneLevel**, la búsqueda está entonces completa y se termina el procedimiento.

   Si el argumento de **subconjunto** es **wholeSubtree**, entonces:

   si el conjunto E del § 2) está vacío, entonces se ha buscado la totalidad del subárbol contenido en este ASG, y, por tanto, terminar el procedimiento;

   en otro caso, continuar el procesamiento de la manera siguiente:

   desígnese por e cada asiento que estaba en el subconjunto E. Repítase el procedimiento de búsqueda del § 2), para cada asiento e.

5) Si los alias han de ser desreferenciados, todo asiento con alias que se encuentre en el paso 2) se inserta en el conjunto D. Para cada asiento d en D, desreferenciar el alias y formular una nueva búsqueda con la **nameResolutionPhase** puesta a **no comenzada**, y con el **targetObject** creado a partir del atributo **aliasedObjectName** y del antiguo nombre del **targetObject**.

   Si el argumento de **subconjunto** era **oneLevel**, se fija a **baseObject**, en la nueva subpetición, y en otro caso fijarlo a **wholeSubtree**.

   Si se devuelve un resultado de error, con motivo de la subpetición ésta deberá ignorarse como si nunca se hubiese hecho.

#### 18.7.2.2.2 *Procedimiento de buscar (II)*

El procedimiento de buscar (II) se aplica a una petición de buscar con el componente **nameResolutionPhase** de **OperationProgress** fijado a **completada**.

El objeto deseado (targetObject) se designará por "e".

Para cada subordinado inmediato contenido localmente e' de e, formular una nueva subpetición con targetObject = e'. Si el argumento de **subconjunto** era **oneLevel**, se fija **baseObject**, en otro caso se deja como **wholeSubtree**. En tal situación, aplicar el procedimiento definido en los pasos 1) a 5) en el § 18.7.2.2.1. Si no hay esos subordinados, retornar el **error de servicio unableToProceed**.

### 18.8 *Procedimiento de fusión de resultados*

Este procedimiento se invoca cuando se está en presencia de resultados externos y/o errores. También puede haber un resultado interno. Se supone que todos los resultados y errores están contenidos en el ASG hasta la conclusión del procedimiento.

La información externa podría deberse a concatenación, difusión o descomposición de petición.

En el caso de concatenación habrá un resultado o error únicos. En el caso de difusión pueden darse tres situaciones: ningún resultado, un resultado, o varios resultados idénticos. Además puede haber algunos errores. Si hay más de un resultado, se retendrá cualquiera de ellos y se descartarán todos los demás. Siempre tiene preferencia el retornar un resultado en vez de un error. Si no hay ningún resultado, se retorna un error, con las excepciones siguientes:

i) si se retornó una **invalidReference**, se marca la referencia como tal, y el ASG puede utilizar una referencia externa alternativa apropiada para continuar la petición, o retornar **ditError** al solicitante; (El tratamiento de las referencias externas no válidas cae fuera del alcance de esta Recomendación.)

ii) en el caso de difusión deben ignorarse los errores **unableToProceed**, salvo si todas las respuestas son de este tipo, en cuyo caso, deberá retornarse al respondedor NameError con **noSuchObject**. Si se retorna al menos un resultado, podrán ignorarse todos los errores;

iii) en el caso de referimientos, éstos no necesitan ser tratados como errores, y puede actuarse con ellos.

Si se requiere la fusión debido a una descomposición de la petición, la fusión produce el mismo efecto que la unión de los resultados.

En el caso de descomposición, cuando han sido fusionados los resultados y los errores, se retorna al solicitante un resultado incompleto.

Es posible que el ASG opte en esta etapa por extraer referimientos de los resultados y errores entrantes que deben estar fusionados. Podría decidir entonces explorar más todos o algunos de ellos, en cuyo caso se concatenarían las operaciones. El antiguo resultado deberá conservarse para fusionarlo luego con los resultados o errores producidos por la concatenación.

El tratamiento de las firmas que pueden figurar en los resultados que se retornen se especifica más adelante en el § 18.9.2.

## 18.9 *Procedimientos para autenticación distribuida*

En esta cláusula se especifican los procedimientos necesarios para prestar los servicios de autenticación distribuida de la guía. Estos servicios, y por tanto esos procedimientos, se dividen en las dos siguientes categorías:

- autenticación de originador, efectuada de forma no protegida (basada en la simple identidad) o en forma segura (basada en firmas digitales); y
- autenticación de resultados, protegidos de forma similar (basada también en firmas digitales).

### 18.9.1 *Autenticación de originador*

#### 18.9.1.1 *Autenticación basada en la identidad*

El servicio de autenticación basado en la identidad permite a los ASG autenticar al solicitante original de la información a los efectos de control de acceso local. Los ASG que deseen explotar este servicio deben seguir el siguiente procedimiento:

- un ASG que deba autenticar una petición PAG obtiene el nombre distinguido del peticionario mediante los procedimientos Vincular, en el momento en que se establece una asociación del AUG (con el ASG). El éxito en la conclusión de estos procedimientos no merma, en manera alguna, el nivel de autenticación que puede exigirse ulteriormente para el tratamiento de operaciones que utilice esa asociación;
- el ASG con el que está asociado el AUG debe insertar el nombre distinguido del solicitante en el campo Iniciador del Argumento de concatenación, en todas las operaciones concatenadas ulteriores, para otros ASG;
- cuando un ASG reciba una operación concatenada, podrá cumplimentarla o no, lo que dependerá de la determinación de los derechos de acceso (un mecanismo definido localmente). Si el resultado no es satisfactorio, puede retornarse un SecurityError con **SecurityProblem** fijado a **insufficientAccessRights**.

#### 18.9.1.2 *Autenticación de originador basada en la firma*

El servicio de autenticación de originador basada en la firma permite a un ASG autenticar (de manera segura) al originador de una determinada petición de servicio. En este punto se describen los procedimientos que sigue un ASG para prestar ese servicio.

El servicio de autenticación basada en firma es invocado por un AUG que utiliza la variante **SIGNED** (firmada) de una petición de servicio firmada facultativamente.

Un ASG que reciba una petición firmada procedente de otro ASG, retirará la firma de este último antes de proceder al tratamiento de la operación. Suponiendo que cualquier verificación de la firma da resultado positivo, el ASG proseguirá la operación. Puede ocurrir que durante el tratamiento, el ASG necesite realizar una concatenación, una difusión o una descomposición de petición, en cuyo caso, el conjunto de argumentos para cada operación concatenada se construirá de la siguiente manera:

- el ASG forma un conjunto de argumentos que pueden ser firmados facultativamente, constituido por el conjunto de argumentos firmados entrante más un **ChainingArgument** modificado.

Si el ASG tiene capacidad de aportar información a la respuesta, puede emplearse autenticación de originador, de acuerdo con la petición de servicio firmada, para determinar los derechos de acceso a esa información.

Cuando un ASG recibe una petición de servicio no firmada, relativa a información que sólo se libera tras la autenticación del originador, se retorna un **SecurityError** con **SecurityProblem** fijado a **protectioRequired.**

18.9.2 *Autenticación de resultados*

Este servicio se proporciona para que los solicitantes de operaciones de guía (sean estos AUG o ASG) puedan verificar (de forma segura utilizando técnicas de firma digital) el origen de los resultados. Es posible pedir el servicio de autenticación de resultados con independencia de si se va a hacer uso o no de la autenticación de originador.

El servicio de autenticación de resultados se inicia empleando el valor **signed** del componente **protectionRequest** tal como figura en el conjunto de argumentos de operaciones de guía: un ASG que reciba una operación, en la que se haya seleccionado esta opción, puede firmar, facultativamente, cualesquiera resultados ulteriores. La opción **signed** en la **protectionRequest,** sirve como indicación al ASG de las preferencias del peticionario; el ASG puede de hecho firmar o no cualesquiera resultados ulteriores.

Si un ASG realiza una concatenación, difusión o descomposición de tal solicitud, tiene varias posibilidades en lo que se refiere a la devolución de resultados al solicitante; a saber:

a) retornarle una respuesta compuesta (firmada o no firmada);

b) retornarle un conjunto de dos o más respuestas parciales no cotejadas (firmadas o no firmadas); en ese conjunto, cero o más miembros pueden estar firmados y cero o uno, no firmados. Si existe un resultado parcial no firmado, este miembro puede ser en realidad un cotejo de una o más respuestas parciales no firmadas, que han sido recibidas de otros ASG, o aportadas por el ASG en cuestión, o de ambos tipos.

## ANEXO A

(a la Recomendación X.518)

## NSA.1 para operaciones distribuidas

Este anexo forma parte de la Recomendación.

Este anexo contiene todas las definiciones de tipo, valor y macro NSA.1 que figuran en esta Recomendación, en forma del módulo NSA.1 **DistributedOperations.**

```
DistributedOperations {joint-iso-ccitt ds(5) modules(1) distributedOperations(3)}
DEFINITIONS ::=
BEGIN
```

```
EXPORTS
        DirectoryRefinement, chainedReadPort, chainedSearchPort, chainedModifyPort,
        DSABind, DSABindArgument,
        DSAUnbind,
        ChainedRead, ChainedCompare, ChainedAbandon,
        ChainedList, ChainedSearch,
        ChainedAddEntry, ChainedRemoveEntry,
        ChainedModifyEntry, ChainedModifyRDN,
        DsaReferral, ContinuationReference;

IMPORTS
        InformationFramework, abstractService, distributedOperations,
        directoryObjectidentifiers, selectedAttributeTypes
            FROM    UsefulDefinitions {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

        DistinguishedName, Name, RelativeDistinguishedName
            FROM    InformationFramework informationFramework

        id-ot-dsa, id-pt-chained-read, id-pt-chained-search, id-pt-chained-modifiy,
            FROM    DistributedDirectoryObjectIdentifiers, distributedDirectoryObjectIdentifiers

        PresentationAddress
            FROM    SelectedAttributeTypes selectedAttributeTypes

        directory, readPort, searchPort, modifyPort
        DirectoryBind,
        ReadArgument, ReadResult,
        CompareArgument, CompareResult,
        Abandon
        ListArgument, ListResult,
        SearchArgument, SearchResult,
        AddEntryArgument, AddEntryResult,
        RemoveEntryArgument, RemoveEntryResult,
        ModifyEntryArgument, ModifyEntryResult,
        ModifyRDNArgument, ModifyRDNResult,
        Abandoned, AttributeError, NameError, ServiceError, SecurityError, UpdateError
        OPTIONALLY-SIGNED, SecurityParameters
            FROM    DirectoryAbstractService directoryAbstractService

-- objetos y puertos --

DirectoryRefinement ::= REFINE directory AS
        dsa         RECURRING
            readPort            [S]   VISIBLE
            searchPort          [S]   VISIBLE
            modifyPort          [S]   VISIBLE
            chainedReadPort           PAIRED WITH dsa
            chainedSearchPort   PAIRED WITH dsa
            chainedModifyPort   PAIRED WITH dsa

dsa OBJECT
            PORTS { readPort            [S],
                    searchPort          [S],
                    modifyPort          [S],
                    chainedReadPort,
                    chainedSearchPort,
                    chainedModifyPort}

        ::= id-ot-dsa

chainedReadPort PORT
        ABSTRACT OPERATIONS {
            ChainedRead, ChainedCompare,
            ChainedAbandon}
        ::= id-pt-chained-read

chainedSearchPort PORT
        ABSTRACT OPERATIONS {
            ChainedList, ChainedSearch}
        ::= id-pt-chained-search
```

```
chainedModifyPort PORT
      ABSTRACT OPERATIONS {
           ChainedAddEntry, ChainedRemoveEntry,
           ChainedModifyEntry, ChainedModifyRDN}
      ::=  id-pt-chained-modify

DSABind   ::=  ABSTRACT-BIND
      TO       {chainedRead,
               chainedSearch,
               chainedModify}
      DirectoryBind

DSAUnbind::=   UNBIND
      FROM     {chainedRead,
               chainedSearch,
               chainedModify}

-- operaciones, argumentos y resultados --

ChainedRead    ::=
      ABSTRACT-OPERATION
           ARGUMENT  OPTIONALLY-SIGNED  SET{
                                        ChainingArgument,
                                        [0]  ReadArgument}
           RESULT    OPTIONALLY-SIGNED  SET{
                                        ChainingResult,
                                        [0]  ReadResult}
           ERRORS {
                DsaReferral, Abandoned, AttributeError, NameError,
                ServiceError, SecurityError}

ChainedCompare      ::=
      ABSTRACT-OPERATION
           ARGUMENT  OPTIONALLY-SIGNED  SET{
                                        ChainingArgument,
                                        [0]  CompareArgument}
           RESULT    OPTIONALLY-SIGNED  SET{
                                        ChainingResult,
                                        [0]  CompareResult}
           ERRORS {
                DsaReferral, Abandoned, AttributeError, NameError,
                ServiceError, SecurityError}

ChainedAbandon      ::=  Abandon

ChainedList    ::=
      ABSTRACT-OPERATION
           ARGUMENT  OPTIONALLY-SIGNED  SET{
                                        ChainingArgument,
                                        [0]  ListArgument}
           RESULT    OPTIONALLY-SIGNED  SET{
                                        ChainingResult,
                                        [0]  ListResult}
           ERRORS {
                DsaReferral, Abandoned, AttributeError, NameError,
                ServiceError, SecurityError}

ChainedSearch  ::=
      ABSTRACT-OPERATION
           ARGUMENT  OPTIONALLY-SIGNED  SET{
                                        ChainingArgument,
                                        [0]  SearchArgument}
           RESULT    OPTIONALLY-SIGNED  SET{
                                        ChainingResult,
                                        [0]  SearchResult}
           ERRORS {
                DsaReferral, Abandoned, AttributeError, NameError,
                ServiceError, SecurityError}
```

```
ChainedAddEntry::=
    ABSTRACT-OPERATION
        ARGUMENT    OPTIONALLY-SIGNED   SET{
                                        ChainingArgument,
                                        [0]   AddEntryArgument}
        RESULT      OPTIONALLY-SIGNED   SET{
                                        ChainingResult,
                                        [0]   AddEntryResult}
        ERRORS {
            DsaReferral, Abandoned, AttributeError, NameError,
            ServiceError, SecurityError, UpdateError}

ChainedRemoveEntry ::=
    ABSTRACT-OPERATION
        ARGUMENT    OPTIONALLY-SIGNED   SET{
                                        ChainingArgument,
                                        [0]   RemoveEntryArgument}
        RESULT      OPTIONALLY-SIGNED   SET{
                                        ChainingResult,
                                        [0]   RemoveEntryResult}
        ERRORS {
            DsaReferral, Abandoned, NameError,
            ServiceError, SecurityError, UpdateError}

ChainedModifyEntry ::=
    ABSTRACT-OPERATION
        ARGUMENT    OPTIONALLY-SIGNED   SET{
                                        ChainingArgument,
                                        [0]   ModifyEntryArgument}
        RESULT      OPTIONALLY-SIGNED   SET{
                                        ChainingResult,
                                        [0]   ModifyEntryResult}
        ERRORS {
            DsaReferral, Abandoned, AttributeError, NameError,
            ServiceError, SecurityError, UpdateError}

ChainedModifyRDN ::=
    ABSTRACT-OPERATION
        ARGUMENT    OPTIONALLY-SIGNED   SET{
                                        ChainingArgument,
                                        [0]   ModifyRDNArgument}
        RESULT      OPTIONALLY-SIGNED   SET{
                                        ChainingResult,
                                        [0]   ModifyRDNResult}
        ERRORS {
            DsaReferral, Abandoned, NameError,
            ServiceError, SecurityError, UpdateError}

-- errores y parámetros --

DSAReferral     ::=
    ABSTRACT-ERROR
        PARAMETER SET {
            [0]   ContinuationReference,
            contextPrefix   [1]   DistinguishedName OPTIONAL}

-- argumentos/resultados comunes --

ChainingArguments   ::=   SET {
    originator              [0]   DistinguishedName OPTIONAL,
    targetObject            [1]   DistinguishedName OPTIONAL,
    operationProgress       [2]   OperationProgress DEFAULT {not Started}
    traceInformation        [3]   TraceInformation,
    aliasdereferenced       [4]   BOOLEAN DEFAULT FALSE,
    AliasedRDNs             [5]   INTEGER OPTIONAL,
    -- ausente a menos que aliasDereferenced es TRUE
```

```
        returnCrossRefs     [6]  BOOLEAN DEFAULT FALSE,
        referenceType       [7]  ReferenceType DEFAULT superior,
        info                [8]  Domaininfo OPTIONAL,
        timeLimit           [9]  UTCTime OPTIONAL,
                            [10] SecurityParameters DEFAULT { }}

ChainingResults     ::=  SET {
        info                [0]  Domaininfo OPTIONAL,
        crossReferences     [1]  SEQUENCE OF CrossReference OPTIONAL,
                            [2]  SecurityParameters DEFAULT { }}

CrossReference      ::=  SET {
        contextPrefix  [0]  DistinguishedName,
        accessPoint         [1]  AccessPoint}

ReferenceType       ::=  ENUMERATED {
              superior                    (1),
              subordinate                 (2),
              cross                       (3),
              nonSpecificSubordinate      (4)}

TraceInformation    ::=  SEQUENCE OF
        SEQUENCE {
              targetObject     Name
              dsa Name,
              OperationProgress}

OperationProgress   ::=  SET {
        NameResolutionPhase       [0]  ENUMERATED {
                                           notStarted(1),
                                           proceeding(2),
                                           completed(3)},
        nextRDNToBeResolved       [1]  INTEGER OPTIONAL}

DomainInfo          ::=  ANY

ContinuationReference     ::=  SET {
        targetObject        [0]  Name,
        aliasedRDNs         [1]  INTEGER OPTIONAL,
        operationProgress   [2]  OperationProgress
        rdnsResolved        [3]  INTEGER OPTIONAL,
        referenceType       [4]  ReferenceType OPTIONAL,
                            -- sólo presente en el DSP --
        accessPoints        [5]  SET OF AccessPoint }

AccessPoint         ::=  SET {
        ae-title       [0]  Name,
        address             [1]  PresentationAddress }
```

## ANEXO B

(a la Recomendación X.518)

### Modelado del conocimiento

Este anexo no forma parte de la Recomendación.

### B.1 *Ejemplo de modelado del conocimiento*

El siguiente ejemplo ilustra la información de conocimiento que deberían mantener los ASG presentados en la figura 5/X.518 (§ 9). En la figura 5/X.518 se representa un AIG dividido lógicamente en cinco contextos de denominación (A, B, C, D y E), distribuido físicamente en tres ASG (ASG1, ASG2 y ASG3). En el ejemplo, el ASG1 contiene el contexto C, el ASG2, los contextos A, B y E, y el ASG3, el contexto D.

En las figuras B-1/X.518 a B-3/X.518 se emplean las siguientes abreviaturas:

SUPR: referencia superior

SUBR: referencia subordinada

INTR: referencia interna

RSNE: referencia subordinada no específica

REFREC: referencia recíproca

ASGn: nombre distinguido del ASGn

PS: dirección de presentación

CP: prefijo de contexto

NDR: nombre distinguido relativo

ASG: nombre distinguido de un ASG

PTR: puntero

NOA: nombre de objeto con alias.

*Nota* - Las figuras siguientes sólo tienen por objeto presentar un ejemplo gráfico de los conceptos definidos en este punto. La manera de almacenar y manejar la información de conocimiento en una implementación concreta de ASG es un asunto local y está fuera del alcance de esta Recomendación.

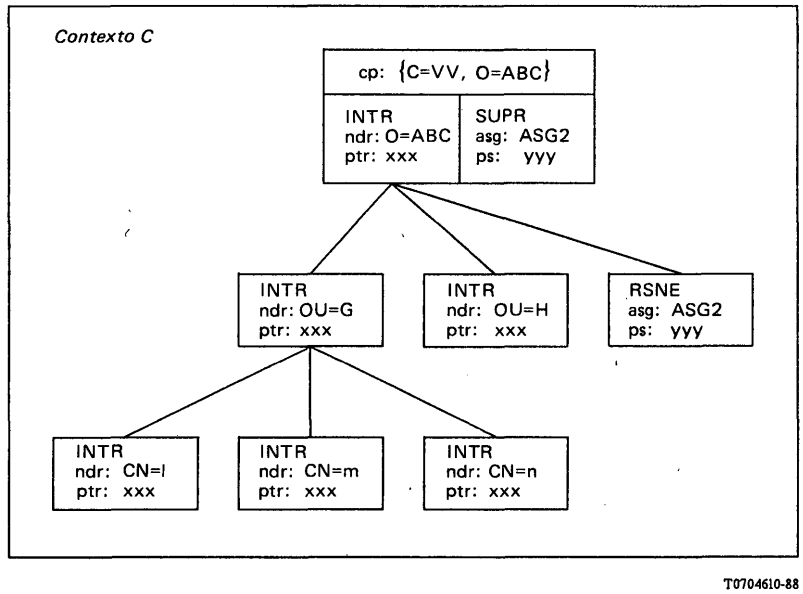



FIGURA B-1/X.518

**Información de conocimiento para el ASG1**

La figura B-1/X.518 ilustra la información de conocimiento que debe contener el ASG1. Ha de incluir los siguientes prefijos de contexto y conjuntos de referencias:

Prefijos de contexto: {C=WW, 0=ABC}, contexto C

Referencias recíprocas: { }

Referencias superiores: {ASG2, dirección de presentación del ASG2}

Referencias internas para el contexto C: {C=WW, O=ABC},
{OU=G}, {OU=H},
{OU=G, CN=1},
{OU=G, CN=m},
{OU=G, CN=n}.

Referencias subordinadas: { }

Referencias subordinadas no específicas: {ASG2, dirección de presentación del ASG2}.

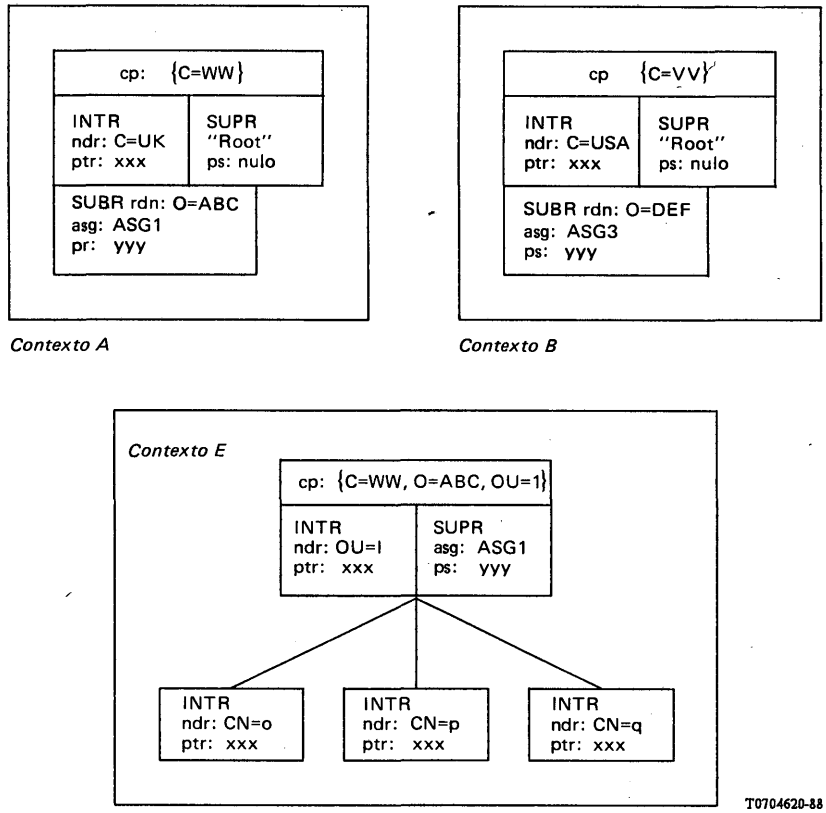



FIGURA B-2/X.518

**Información de conocimiento para el ASG2**

La figura B-2/X.518 ilustra la información de conocimiento que debe contener el ASG2. Ha de incluir los siguientes prefijos de contexto y conjuntos de referencias:

Prefijos de contextos: {C=WW}, contexto A
{C=VV}, contexto B
{C=WW, O=ABC, OU=I}, contexto E

Referencias recíprocas: { }

Referencias superiores: { }

Referencias internas para el contexto A: {C=WW}

Referencias internas para el contexto B: {C=VV}

Referencias internas para el contexto E: {C=WW, O=ABC, OU=I},
{CN=o},
{CN=p},
{CN=q}.

Referencias subordinadas para el contexto A: {C=WW, O=ABC}

Referencias subordinadas para el contexto B: {C=VV, O=DEF}

Referencias subordinadas no específicas: { }

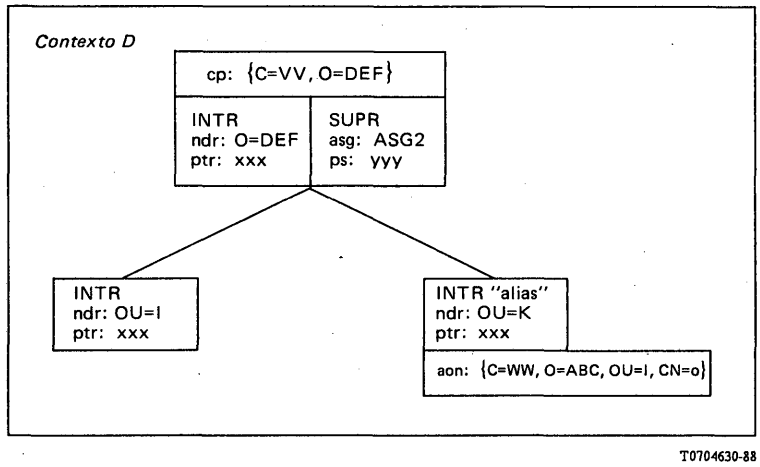



FIGURA B-3/X.518

**Información de conocimiento para el ASG3**

La figura B-3/X.518 ilustra la información de conocimiento que debe contener el ASG3. Ha de incluir los siguientes prefijos de contexto y conjuntos de referencias:

| | |
|---|---|
| Prefijos de contexto: | {C=VV, O=DEF}, contexto D |
| Referencias recíprocas | {{C=WW, O=ABC, OU=H}, ASG1, dirección de presentación de ASG1 (no se muestra en la figura anterior) |
| Referencias superiores: | {ASG2, dirección de presentación de ASG2} |
| Referencias internas para el contexto D: | {ASG1, dirección de presentación de ASG1}<br>{C=VV, O=DEF},<br>{OU=J},<br>{OU=K} alias para {C=WW, O=ABC, OU=I, CN=o}<br>(la información de alias no forma parte del conocimiento) |
| Referencias subordinadas: | { } |
| Referencias subordinadas no específicas | { } |

## B.2 *Ejemplo de resolución de nombre distribuido*

Lo que sigue es un ejemplo de cómo se utiliza la resolución de nombre distribuido para tratar diferentes peticiones de guía. El ejemplo se basa en el AIG hipotético que se muestra en la figura 5/X.518 (§ 9) y en las correspondientes configuraciones de ASG de las figuras B-1/X.518 a B-3/X.518 (anexo B).

En la hipótesis de modo de propagación por concatenación, las siguientes peticiones, dirigidas al ASG1, se tratarían como se indica:

1) Una petición con nombre distinguido {C=WW, O=ABC, OU=G, CN=1}

   - Concordará con el prefijo de contexto {C=WW, O=ABC} del contexto C para el que el ASG1 tiene autoridad administrativa. Por ello, la resolución de nombre comenzará en el ASG1 con contexto C.
   - La resolución de nombre avanzará en sentido descendente en el contexto C concordando cada uno de los NDR restantes hasta que se localice CN=1.

2) Una petición con nombre distinguido {C=WW, O=JPR}

- No concordará con ningún prefijo de contexto contenido en el ASG1, por lo que el ASG1 utilizará su referencia superior para transferir la petición al ASG2 y al ASG3, superiores a él.
- En el ASG2 la petición concordará con el prefijo de contexto {C=WW} y comenzará la resolución de nombre con contexto A.
- La resolución de nombre no encontrará un subordinado de C=WW con el que concordar el NDR O=JPR, por lo que la petición resultará fallida y se determinará que el nombre era válido (es decir, que se refería a un objeto no existente).

3) Una consulta con nombre distinguido {C=VV, O=DEF, OU=K}

- No concordará con ningún prefijo de contexto contenido en el ASG1.
- El ASG1 transferirá, por tanto, la petición a su ASG superior, el ASG2.
- La petición concordará con el prefijo de contexto {C=VV} del contexto B contenido en el ASG2. Por ello, la resolución de nombre comenzará en el ASG2 con contexto B.
- Al tratar el procedimiento de resolución de nombre de concordar O=DEF encontrará una referencia subordinada indicando que {C=VV, O=DEF} es el comienzo de un nuevo contexto contenido el ASG3.
- La resolución de nombre proseguirá en el ASG3 hasta que se localice {C=VV, O=DEF, CN=K}.
- En el supuesto de que haya que desreferenciar alias, se elaborará un nombre nuevo utilizando el nombre con alias contenido en el asiento {C=VV, O=DEF, CN=K}. El nombre nuevo resultante será: {C=WW, O=ABC, OU=I, CN=o}.
- El ASG3 reanudará el tratamiento de la solicitud utilizando el nombre nuevo obtenido por desreferenciación.

## ANEXO C

(a la Recomendación X.518)

## Uso distribuido de la autenticación

Este anexo no forma parte de la Recomendación.

### C.1 *Resumen*

El modelo de seguridad se define en el § 10 de la Recomendación X.501. A continuación se resumen los puntos principales del modelo.

a) La autenticación simple del iniciador de la operación no es soportada en el protocolo de sistema de guía (PSG).

b) La autenticación fuerte, mediante la firma de la petición y del resultado, no es soportada en el PSG.

c) La encripción de la petición o de los resultados, no es soportada en el PSG.

d) La autenticación de errores, incluidos los referimientos, no es soportada en el PSG.

Este anexo describe la forma en que se realiza lo indicado en b) en la guía distribuida. En él se utilizan la terminología y la notación definidas en la Recomendación X.509.

### C.2 *Autenticación simple*

El AUG es autenticado como parte de la operación Vincular del Protocolo de Acceso a la Guía (PAG). De allí en adelante, en el PSG sólo se lleva el nombre del AUG en el campo Iniciador del Argumento de Concatenación.

C.3 *Modelo de autenticación distribuida*

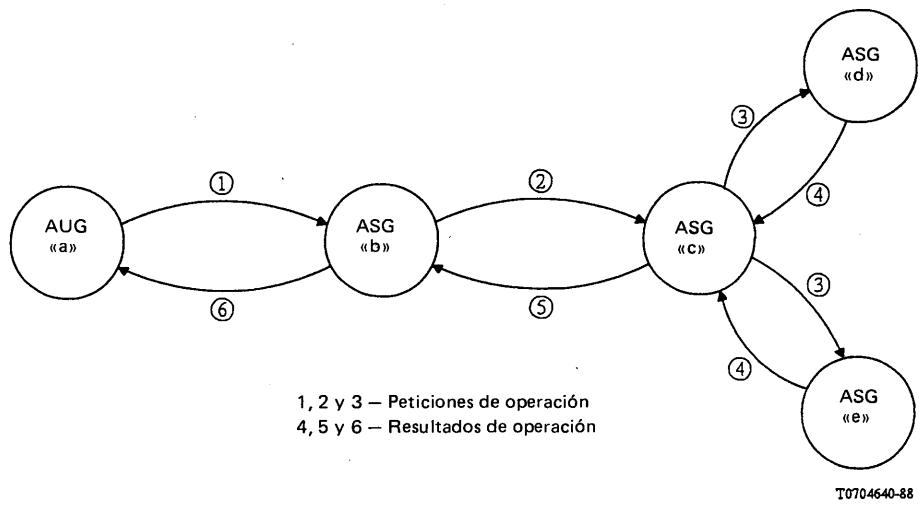



FIGURA C-1/X.518

**Modelo de autenticación distribuida**

En la figura C-1/X.518 se indica el modelo que debe utilizarse para especificar los procedimientos de autenticación distribuida. El modelo identifica la secuencia de flujos de información para el caso general de una operación de buscar o listar. Se considera que la operación se origina en el AUG "a" y en ella se menciona un objeto buscado que se halla en el ASG "c". En la ejecución de la operación participan los ASG "b", "c", "d" y "e".

El AUG "a" se pone en contacto inicialmente con un ASG cualquiera (el ASG "b") que no contiene el objeto deseado pero que puede navegar, mediante concatenación, hacia el ASG que si lo contiene (el ASG "c"). Si todos los ASG estuvieran funcionando en modo referimiento, el modelo se simplificaría notablemente y cada intercambio AUG/ASG equivaldría, desde el punto de vista de la autenticación a la interacción entre el AUG "a" y el ASG "b".

C.4 *AUG a ASG*

La autenticación del originador se efectúa como consecuencia del intercambio (1) de la figura C-1/X.518. El procedimiento de autenticación es el siguiente:

Siendo

OA = el argumento de la operación, es decir, el argumento de buscar, leer, comparar, etc., tal como se definen en la parte 3.

y

a{OA} = el argumento de la operación firmado por el AUG "a".

La autenticidad se determina verificando la firma.

C.5 *Transferencia del PAG al PSG*

Este procedimiento es el que realiza el ASG "b" de la figura C-1/X.518 y representa la transferencia de la identidad firmada del iniciador del PAG al PSG.

El ASG "b" formula el argumento de concatenación adecuado tal como se describe en el § 12.3 de esta Recomendación y la combina con el argumento de la operación del PAG formando así una operación concatenada o sea, concatenado-leer, concatenado-buscar, concatenado-listar, etc. del PSG. La operación concatenada así constituida será firmada antes de pasarla a otro ASG (el ASG "c" de la figura C-1/X.518). La estructura de datos puede representarse de la siguiente forma:

b{ChA, a{OA}} = la operación concatenada firmada por el ASG "b"

en donde

ChA = argumento de concatenación.

La información de autenticación transportada en el PSG entre dos ASG [intercambio (2) en la figura C-1/X.518] consta, por tanto, de dos partes:

- el argumento de operación, firmado por el iniciador, que permite la autenticación del iniciador;
- la operación concatenada, firmada por el ASG emisor, que permite la autenticación del ASG emisor.

C.6. *Concatenación a través de ASG intermedios*

Este procedimiento sería realizado por el ASG "c" en el modelo representado en la figura C-1/X.518. El ASG "c" descarta la firma suministrada por el ASG emisor (ASG "b" de la figura C-1/X.518) y modifica el argumento de concatenación tal como se describe en el § 12.3 de la presente Recomendación. A continuación combina el argumento de concatenación modificado con el argumento de operación firmado y firma el resultado creando una operación concatenada firmada modificada. Esto se puede representar por:

c{ChA', a{OA}} = la operación concatenada firmada por el ASG "c"

en donde

ChA' = argumento de concatenación modificado.

La operación concatenada modificada se representa en la figura C-1/X.518 por el intercambio (3). Según la naturaleza de la operación y el tipo de conocimiento contenido, el ASG "c" puede realizar una descomposición de la consulta antes de concatenar o difundir cualquier operación (cualesquiera operaciones) resultante(s). Esto es lo que se representa en la figura C-1/X.518 con el envío, por parte del ASG "c", de operaciones al ASG "d" y al ASG "e". En cada caso el procedimiento de autenticación es el mismo.

C.7 *Autenticación de resultados*

Un iniciador de operación de guía solicita el servicio de autenticación de resultados utilizando la opción **signed** en el **securityParameter protectionRequest**. Al contestar a esa solicitud, un ASG puede decidir facultativamente si firma o no algunos o todos los resultados. El servicio de autenticación de resultados no facilita la autenticación de respuestas de error.

En el contexto de un ASG determinado que trata los resultados procedentes de un número cualquiera de ASG (cada uno de los cuales está asociado a una petición de servicio particular), son posibles estos tres distintos casos:

- el ASG proporciona un conjunto completo de resultados de una operación sin necesidad de ninguna función de cotejo (caso que representan el ASG "d" y el ASG "e" de la figura C-1/X.518);
- el ASG coteja los resultados locales (emanados de este mismo ASG) con los de otro u otros ASG (caso representado por el ASG "c" de la figura C-1/X.518);
- el ASG concatena el resultado de un ASG a otro ASG o a un AUG y, sin contribuir por ello al conjunto de resultados (caso del ASG "b" de la figura C-1/X.518).

C.7.1 *Resultados de ASG - Sin cotejo*

En este punto se analiza el papel desempeñado por un ASG que sea la única fuente de resultados para una petición de operación determinada, es decir, que no tenga que efectuar ninguna función de cotejo. Este caso se examina tanto para el PSG como para el PAG.

C.7.1.1 *PSG*

El ASG puede elegir la realización de alguno de los siguientes procedimientos:

- retornar los resultados no firmados, lo que puede representarse por:

  ChR, OR = resultado de la operación concatenada (no firmado)

  en donde

  ChR = resultados de la concatenación

  OR = resultado de la operación;
- firmar solamente el resultado de la operación, lo que puede representarse por:

  ChR, d(OR) = resultado de la operación concatenada firmado por el ASG "d";
- firmar solamente el resultado de la operación concatenada, lo que puede representarse por:

d(ChR, OR) = resultado de la operación concatenada firmado por el ASG "d";

- firmar tanto el resultado de la operación como el de la operación concatenada, lo que puede representarse por:

  d{ChR, D{OR}} = resultado de la operación y resultado de la operación concatenada, firmados por el ASG "d".

*Nota* - Si el resultado de la operación está firmado, se remitirá ese resultado firmado al iniciador. Si se ha firmado el resultado de la operación concatenada, el ASG receptor deberá desechar la firma para modificar el argumento de resultados concatenados antes de transferir el resultado de la operación concatenada.

C.7.1.2 *PAG*

El PAG se describe en forma completa en la Recomendación X.511. Aquí sólo se reproduce un resumen de esa descripción a efectos de la integridad de la exposición.

El ASG puede elegir entre retornar los resultados no firmados, que pueden representarse por:

OR = resultado de la operación;

o retornarlos firmados, lo que puede representarse por:

d{OR} = resultado de la operación firmado por el ASG "d".

C.7.2 *Resultados de ASG - Con cotejo*

En esta cláusula se analiza el papel desempeñado por un ASG que retorna el resultado de una petición de servicio determinada cuando sea requisito previo el cotejo e integración de los resultados de otros ASG. Este caso se examina tanto para el PSG como para el PAG.

C.7.2.1 *PSG*

Habida cuenta de que pueden estar firmados cero o más resultados recibidos de otros ASG, este procedimeinto permite a un ASG cotejar e integrar los resultados y firmar cero o más partes componentes de un resultado compuesto y, facultativamente, firmar el propio resultado compuesto como un todo.

C.7.2.1.1 *Producción del argumento de resultados de concatenación*

Este procedimiento exige que un ASG (representado por el ASG "c" en la figura C-1/X.518) elimine todas las firmas de resultado de operación concatenada de los resultados recibidos de ASG externos (el ASG "d" y el ASG "e" en la figura C-1/X.518). El ASG "c" posee entonces de un conjunto de resultados de concatenación no firmados, un conjunto de resultados de operación firmados y un conjunto de resultados de operación no firmados.

Todos los resultados de concatenación son manipulados, tal como se describe en el § 12.4 de esta Recomendación, para crear un resultado de concatenación modificado único designado por:

i) ChR' = resultados de concatenación modificados.

C.7.2.1.2 *Resultado obtenido localmente no firmado*

Si el ASG no desea firmar los resultados generados localmente, el conjunto de resultados de operaciones no firmadas se fusiona antes con el resultado local para formar un conjunto modificado de resultados de operaciones designado por:

OR' = resultado de operación fusionado.

El conjunto completo de resultados de operación está entonces constituido por la unión del conjunto de resultados de operación, firmados externamente, representado por:

d{OR}, e{OR} ...

y el resultado de la operación fusionado, unión a la que se representa de forma colectiva por:

ii) OR', d{OR}, e{OR} ... = resultado de operación.

C.7.2.1.3 *Resultado obtenido localmente firmado*

Si el ASG desea firmar los resultados generados localmente, se fusiona en primer lugar el conjunto generado externamente de resultados de operación no firmados. El conjunto completo de resultados de operaciones es, entonces, la unión del conjunto firmado localmente de resultados de

operación designado por C{OR}, el conjunto fusionado de resultados de operación no firmados designado por OR", y el conjunto de resultados de operación firmados externamente, designado por:

d{OR, e{OR}, ..., unión que se representa de forma colectiva por:

iii) c{OR}, OR", d{OR}, e{OR}, ... = resultado de operación.

C.7.2.1.4 *Resultado de operación concatenada no firmado*

Si el ASG no desea firmar el resultado de la operación concatenada, éste constará de los resultados de la concatenación [identificados en i)] junto con el resultado de la operación identificado en ii) o en iii), agrupación que, de forma colectiva, se representa ya sea por:

ChR', OR', d{OR}, e{OR}, ... = resultado de operación concatenada (no firmado).

o bien por:

ChR', c{OR}, OR", d{OR}, e{OR}, ... = resultado de operación concatenada (no firmado) y resultado de operación firmado por el ASG "c".

C.7.2.1.5 *Resultado de operación concatenada firmado*

Si el ASG no desea firmar el resultado de la operación concatenada, éste constará de los resultados de la concatenación [identificados en i)] junto con el resultado de la operación identificado en ii) o en iii), agrupación que, de forma colectiva, se representa ya sea por:

c{ChR', OR', d{OR}, e{OR}, ...} = resultado de operación concatenada firmado por el ASG "c".

o,

c{ChR', c{OR}, OR", d{OR}, e{OR}, ...} = resultado de operación concatenada y resultado de operación firmados por el ASG "c".

C.7.2.2 *En el PAG*

El procedimiento es muy similar al descrito en el § C.7.2.1, del que sólo difiere en que en el PAG no se pasa el argumento de resultados de concatenación.

C.7.3 *Resultados concatenados por ASG*

En este punto se analizan los procedimientos que debe seguir un ASG para concatenar un resultado de operación devolviéndolo al solicitante de la misma, ASG o AUG, en el PSG y en el PAG, respectivamente.

C.7.3.1 *PSG*

El ASG retira, en primer lugar, la firma (si existe alguna) del resultado de la operación concatenada. A continuación, manipula el argumento de resultados de concatenación como se describe en la presente Recomendación para producir un argumento de resultados de concatenación modificado. Este se fusiona de nuevo con el argumento de resultado de operación para producir un resultado de operación concatenada modificado. Por último, el ASG puede, si lo desea, firmar el resultado de operación concatenada antes de pasarlo al siguinte ASG de la cadena.

C.7.3.2 *PAG*

Un ASG (el ASG "b" en el caso de la figura C-1/X.518) retira, en primer lugar, la firma (si existe alguna) del resultado de operación concatenada. A continuación analiza y descarta el argumento de resultados de concatenación y, finalmente, firma, si lo desea, el argumento de resultado de operación restante antes de pasar el resultado de AUG.

## ANEXO D

(a la Recomendación X.518)

**Identificadores de objeto de guía distribuida**

Este anexo forma parte de la Recomendación.

Este anexo contiene todos los identificadores de objetos NSA.1 que figuran en esta Recomendación, en forma del módulo NSA.1 **DistributedDirectoryObjectIdentifiers.**

```
DistributedDirectoryObjectIdentifiers      {joint-iso-ccitt ds(5) modules(1)
                                            distributedDirectoryObjectIdentifiers(13)}

DEFINITION ::=
BEGIN

EXPORTS
      id-ot-dsa, id-pt-chainedRead, id-pt-chainedSearch, id-pt-chainedModify;

IMPORTS
      id-ot, id-pt
            FROM    UsefulDefinitions    {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)};

-- objetos --

id-ot-dsa   OBJECT IDENTIFIER    ::=   {id-ot 3}

-- tipos de puertos --

id-pt-chainedRead    OBJECT IDENTIFIER    ::=   {id-pt 4}
id-pt-chainedSearch  OBJECT IDENTIFIER    ::=   {id-pt 5}
id-pt-chainedModify  OBJECT IDENTIFIER    ::=   {id-pt 6}

END
```

**Recomendación X.519**

## LA GUIA - ESPECIFICACIONES DE PROTOCOLOS [1)]

*(Melbourne, 1988)*

INDICE



---

1) La Recomendación X.519 y la norma ISO 9594-5, The Directory-Protocol Specifications (La guía - Especificaciones de protocolos) se redactaron en estrecha colaboración y están técnicamente alineadas.

## 0 Introducción

0.1 La finalidad de este documento, y de los otros de la misma serie, es facilitar la interconexión de sistemas de procesamiento de información para la prestación de servicios de guía. El conjunto de todos estos sistemas, junto con la información contenida en la guía, puede considerarse como un todo denominado la *guía*. La información contenida en la guía, conocida colectivamente como la base de información de la guía (BIG), se utiliza típicamente para facilitar la comunicación entre objetos, con ellos o respecto a ellos; por ejemplo, terminales, personas, entidades de aplicación y listas de distribución.

0.2 La guía desempeña un papel importante en la interconexión de sistemas abiertos, cuyo propósito es permitir con un mínimo de acuerdos técnicos fuera de las normas de interconexión mismas, la interconexión de sistemas de procesamiento de información:

- de diferentes fabricantes;
- sometidos a gestiones diferentes;
- de diferentes grados de complejidad; y
- de diferentes fechas de construcción.

0.3 Esta Recomendación especifica los elementos de servicio de aplicación y contextos de aplicación para dos protocolos - el protocolo de acceso a la guía (PAG) y el protocolo de sistema de guía (PSG). El PAG proporciona el acceso a la guía para extraer o modificar información de la guía. El PSG proporciona la concatenación de indagaciones (interrogaciones, consultas) para extraer o modificar información de la guía que está contenida en otras partes de un sistema de guía distribuida.

## 1 Alcance

Esta Recomendación especifica el protocolo de acceso a la guía y el protocolo de sistema de guía, de conformidad con los servicios abstractos especificados en las Recomendaciones X.511 y X.518.

## 2 Referencias

Recomendación X.200 - Modelo de referencia de interconexión de sistemas abiertos para aplicaciones del CCITT

Recomendación X.208 - Especificación de la notación de sintaxis abstracta uno (NSA.1)

Recomendación X.209 - Especificación de las reglas básicas de codificación de notación de sintaxis abstracta uno (NSA.1)

Recomendación X.500 - La guía - Visión de conjunto de conceptos, modelos y servicios

Recomendación X.501 - La guía - Modelos

Recomendación X.511 - La guía - Definición del servicio abstracto

Recomendación X.518 - La guía - Procedimientos para operación distribuida

Recomendación X.520 - La guía - Tipos de atributo seleccionados

Recomendación X.521 - La guía - Clases de objeto seleccionadas

Recomendación X.219 - Operaciones a distancia: modelo, notación y definición del servicio

Recomendación X.229 - Operaciones a distancia: especificación de protocolo

Recomendación X.217 - Definición del servicio de control de asociación para la interconexión de sistemas abiertos para aplicaciones del CCITT

Recomendación X.227 - Especificación del protocolo de control de asociación para la interconexión de sistemas abiertos

Recomendación X.216 - Definición del servicio de presentación para la interconexión de sistemas abiertos para aplicaciones del CCITT

## 3 Definiciones

En las definiciones contenidas en este punto se utilizan las abreviaturas definidas en el § 4.

### 3.1 *Definiciones relativas al modelo de referencia ISA*

En esta Recomendación, que se basa en los conceptos desarrollados en la Recomendación X.200, se utilizan los siguientes términos definidos en la misma:

a) *elemento del servicio de aplicación*;

b) *información de control de protocolo de aplicación*;

c) *unidad de datos de protocolo de aplicación*;

d) *contexto de aplicación*;

e) *entidad de aplicación*;

f) *sintaxis abstracta.*

### 3.2 *Definiciones básicas relativas a la guía*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.501:

a) *la guía*;

b) *usuario (de la guía)*;

c) *agente de sistema de guía (ASG)*;

d) *agente de usuario de la guía (AUG).*

### 3.3 *Definiciones relativas a las operaciones distribuidas*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.518:

a) *concatenación*;

b) *referimiento.*

## 4 Abreviaturas

En esta Recomendación se utilizan las siguientes abreviaturas:

ASG Agente de sistema de guía
AUG Agente de usuario de guía
CA Contexto de aplicación
EA Entidad de aplicación
ESA Elemento de servicio de aplicación
ESCA Elemento de servicio de control de asociación
ESOD Elemento de servicio de operaciones distantes
ICPA Información de control de protocolo de aplicación
PAG Protocolo de acceso a la guía
PSG Protocolo de sistema de guía
UDPA Unidad de datos de protocolo de aplicación

## 5 Convenios

En la Recomendación se utilizan los siguientes convenios:

a) las definiciones de sintaxis abstracta en el § 7 se definen utilizando la notación de sintaxis abstracta definida en la Recomendación X.208;

b) las macros de operación distante (notación OD) y las macros de elemento de servicio de aplicación y de contexto de aplicación se definen en la Recomendación X.219;

c) las palabras que forman términos definidos y los nombres y valores de parámetros de servicio y de campos de protocolo, si no son nombres propios, comienzan con minúscula y van unidos por un guión, por ejemplo, término-definido. Los nombres propios comienzan con mayúscula y no van unidos con guión, por ejemplo: Nombre Propio.

## 6 Visión de conjunto de los protocolos

### 6.1 *Modelo de protocolo de guía*

La Recomendación X.511 define el servicio abstracto entre un AUG y la guía para soportar el acceso del usuario a los servicios de la guía. La guía se modela también representada por un ASG que soporta el punto de acceso en cuestión. La Recomendación X.518 define las interacciones entre un par de ASG dentro de la guía para soportar peticiones de usuario que están concatenadas. Estos conceptos se ilustran en la figura 1/X.519.

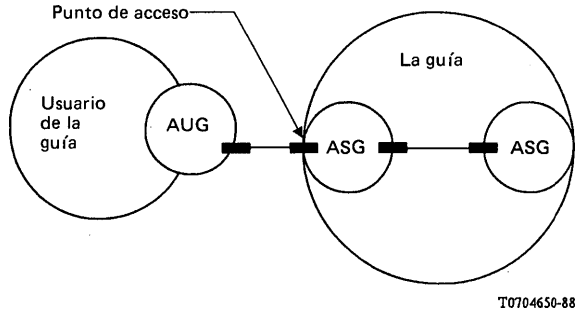



FIGURA 1/X.519

**Interacciones de la guía**

Cuando un AUG está en un sistema abierto diferente del ASG con el cual está interactuando, estas interacciones son soportadas por el protocolo de acceso a la guía (PAG), que es un protocolo de la capa de aplicación ISA. Igualmente, cuando dos ASG que están interactuando se encuentran en diferentes sistemas abiertos, las interacciones son soportadas por el protocolo de sistema de guía (PSG), que es también un protocolo de la capa de aplicación.

Tanto el PAG como el PSG son protocolos que proporcionan comunicación entre un par de procesos de aplicación. En el entorno ISA esto se representa como una comunicación entre un par de entidades de aplicación (EA) que utilizan el servicio de presentación. La función de una EA la proporciona un conjunto de elementos de servicio de aplicación (ESA). La interacción entre las EA se describe atendiendo a la utilización, por las EA, de los servicios proporcionados por los ESA. Los dos ESA comunes a ambos protocolos de la guía se resumen en este punto.

El elemento de servicio operaciones distantes (ESOD) soporta el paradigma petición/respuesta de la operación abstracta que tiene lugar en los puertos, en el modelo abstracto. Los ESA de la guía proporcionan la función de correspondencia de la notación de sintaxis abstracta del servicio abstracto de guía con servicios proporcionados por el ESOD.

El elemento de servicio control de asociación (ESCA) soporta el establecimiento y la liberación de una asociación de aplicación entre un par de EA. Las asociaciones entre un AUG y un ASG sólo puede establecerlas el AUG. Una asociación establecida sólo puede ser liberada por el iniciador.

### 6.2 *Protocolo de acceso a la guía*

El protocolo de acceso a la guía (PAG) se utiliza para realizar el servicio abstracto de guía. Comprende, además de ESOD y ESCA, tres ESA específicos de la guía: **readASE, searchASE** y **modifyASE**. Dichos elementos corresponden a **readPort, searchPort** y **modifyPort** del servicio abstracto. El contexto de aplicación **directoryAccessAC** identifica la combinación de: **readASE, searchASE** y **modifyASE aCSE, rOSE.**

### 6.3 *Protocolo de sistema de guía*

El protocolo de sistema de guía (PSG) se utiliza para realizar la funcionalidad de la operación distribuida descrita en la Recomendación X.518. Comprende, además de ESOD (ROSE) y ESCA (ACSE), tres ESA específicos de la guía: **chainedReadASE, chainedSearchASE,** y **chainedModifyASE,** que corresponden a **chainedReadPort, chainedSearchPort** y **chainedModifyPort,** del servicio abstracto. El contexto de aplicación **DirectorySystemAC** identifica la combinación de: **chainedReadASE, chainedSearchASE,** y **chainedModifyASE, aCSE, rOSE.**

### 6.4 *Utilización de servicios subyacentes*

Los protocolos PAG y PSG utilizan los servicios subyacentes que se describen a continuación.

#### 6.4.1 *Utilización de servicios ESOD*

El elemento de servicio de operaciones distantes (ESOD) se define en la Recomendación X.219.

El ESOD soporta el paradigma petición/respuesta de operaciones distantes.

Los ESA de la guía son usuarios de los servicios OD-INVOCACION, OD-RESULTADO, OD-ERROR, OD-RECHAZAR-U y OD-RECHAZAR-P, del ESOD.

Las operaciones distantes del PAG y del PSG son operaciones de clase 2 (asíncronas). Obsérvese que el ASG es un consumidor del PAG y puede decidir operar de un modo síncrono.

El PAG utiliza asociación de clase 1. Esto significa que el ASG no puede invocar operaciones sobre el AUG. El PSG utiliza asociación de clase 3. Esto significa que el ASG respondedor puede invocar operaciones sobre el ASG iniciador, y viceversa.

#### 6.4.2 *Utilización de servicios ESCA*

El elemento de servicio control de asociación (ESCA) se define en la Recomendación X.217.

El ESCA proporciona el control (establecimiento, liberación, aborto) de asociaciones de aplicación entre EA.

Vincular a la guía y Desvincular de la guía (o Vincular ASG y Desvincular ASG) son los únicos usuarios de los servicios A-ASOCIACION y A-LIBERACION del ESCA en modo normal. El proceso de aplicación es el usuario de los servicios A-ABORTO y A-P-ABORTO del ESCA.

6.4.3 *Utilización del servicio de presentación*

El servicio de presentación se define en la Recomendación X.216.

La capa de presentación coordina la representación (sintaxis) de la semántica de la capa de aplicación que ha de intercambiarse.

En el modo normal se utiliza un contexto de presentación diferente para cada sintaxis abstracta incluida en el contexto de aplicación.

El ESCA es el único usuario de los servicios P-CONEXION, P-LIBERACION, P-U-ABORTO y P-P-ABORTO del servicio de presentación.

El ESOD es un usuario del servicio P-DATOS del servicio de presentación.

6.4.4 *Utilización de servicios de capa inferior*

El servicio de sesión se define en la Recomendación X.215. La capa de sesión estructura el diálogo del flujo de información entre los sistemas de extremo.

Las unidades funcionales Kernel y Dúplex del servicio de sesión son utilizadas por la capa de presentación.

El servicio de transporte se define en la Recomendación X.214. La capa de transporte proporciona la transferencia transparente de extremo a extremo de datos a través de la conexión de red subyacente.

La elección de la clase del servicio de transporte utilizado por la capa de sesión depende de las exigencias de multiplexación y recuperación tras error. El soporte de la clase de transporte 0 (sin multiplexación) es obligatorio. El servicio de transporte acelerado no se utiliza.

El soporte de otras clases es facultativo. Puede utilizarse una clase de multiplexación para multiplexar el PAG o el PSG, y otros protocolos en la misma conexión de red. Se puede elegir una clase de recuperación tras error para una conexión de red con una tasa aceptable de errores residuales.

Se supone que una red subyacente soporta el servicio de red ISA definido en la Recomendación X.213.

La dirección de red se define en la Recomendación X.121, en las Recomendaciones E.163/E.164 o en la Recomendación X.200 (dirección PASR para la ISA).

## 7 Sintaxis abstracta de protocolo de la guía

7.1 *Sintaxis abstractas*

Los ESA de la guía especificados en los § 7.2.1, 7.2.3 y 7.2.5 utilizan una sola sintaxis abstracta, **id-as-directory-AccessAS**. Los especificados en los § 7.2.2, 7.2.4 y 7.2.6 utilizan también una sola sintaxis abstracta, **id-as-directorySystemsAS**. Cada una de ellas define la información de control de protocolo de aplicación (ICPA) que, cuando se utiliza junto con ESOD, define un conjunto de UDPA. Las UDPA de la guía se definen en la sintaxis abstracta de los ESA y ESOD de la guía. Estos, junto con la sintaxis abstracta de ESCA forman la definición completa de UDPA utilizadas durante una asociación de la guía.

La sintaxis abstracta ESCA **id-as-acse** se necesita para establecer las asociaciones.

Estas sintaxis abstractas serán (como mínimo) codificadas de acuerdo con las reglas básicas de codificación de NSA.1.

7.2 *Elementos de servicio de aplicación de la guía*

Esta cláusula especifica los ESA que se utilizan como "bloques constructivos" en la formación de diversos contextos de aplicación de la guía en el § 7.3.

*Nota* - Estos ESA se utilizan para la construcción de los conceptos de aplicación definidos en esta Recomendación. No tienen por objeto permitir que se anuncie la conformidad con cada uno de los distintos ESA, ni con una combinación de ellos.

7.2.1 *Leer ESA (read ASE)*

El **readASE** soporta las operaciones del **readPort**, es decir, **Read**, **Compare** y **Abandon**, definidas en la Recomendación X.511.

```
readASE
        APPLICATION-SERVICE-ELEMENT
                CONSUMER INVOKES
                        {read, compare, abandon}
        ::=     id-ase-readASE

read            Read            ::= 1

compare         Compare         ::= 2

abandon         Abandon         ::= 3
```

7.2.2 *Concatenado Leer ESA (chained Read ASE)*

El **chainedReadASE** soporta las operaciones del **ChainedReadPort**, es decir, **ChainedRead**, **ChainedCompare** y **ChainedAbandon**, definidas en la Recomendación X.518.

```
chainedReadASE
        APPLICATION-SERVICE-ELEMENT
                OPERATIONS {
                        chainedRead,
                        chainedCompare,
                        chainedAbandon}
        ::=     id-ase-chainedReadASE

chainedRead             ChainedRead             ::= 1

chainedCompare          ChainedCompare          ::= 2

chainedAbandon          ChainedAbandon          ::= 3
```

7.2.3 *Buscar ESA (search ASE)*

El **searchASE** soporta las operaciones abstractas de **SearchPort**, es decir, **List** y **Search**, definidas en la Recomendación X.511.

```
searchASE
        APPLICATION-SERVICE-ELEMENT
                CONSUMER INVOKES { list, search}
        ::=     id-ase-searchASE}

list            List            ::= 4

search          Search          ::= 5
```

7.2.4 *Concatenado Buscar ESA (chained Search ASE)*

El **chainedSearchASE** soporta las operaciones abstractas del **ChainedSearchPort**, es decir, **ChainedList** y **ChainedSearch**, definidas en la Recomendación X.518.

```
chainedSearchASE
        APPLICATION-SERVICE-ELEMENT
                OPERATIONS {
                        chainedList, chainedSearch}
        ::=     id-ase-chainedSearchASE

chainedList             ChainedList     ::= 4

chainedSearch           ChainedSearch   ::= 5
```

7.2.5 *Modificar ESA (modify ASE)*

El **modifyASE** soporta las operaciones abstractas del **ModifyPort**, es decir, **addEntry**, **RemoveEntry**, **ModifyEntry** y **ModifyRDN**, definidas en la Recomendación X.511.

```
modifyASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
            {addEntry, removeEntry,
            modifyEntry, modifyRDN}
    ::=  id-ase-modifyASE

addEntry        AddEntry        ::= 6
removeEntry     RemoveEntry     ::= 7
modifyEntry     ModifyEntry     ::= 8
modifyRDN       ModifyRDN       ::= 9
```

7.2.6 *Concatenado Modificar ESA (chained Modify ASE)*

El **chainedModifyASE** soporta las operaciones abstractas del **ChainedModifyPort**, es decir, **ChainedAddEntry**, **ChainedRemoveEntry**, **ChainedModifyEntry**, y **ChainedModifyRDN**, definidas en la Recomendación X.518.

```
chainedModifyASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS
            {chainedAddEntry,
            chainedRemoveEntry,
            chainedModifyEntry,
            chainedModifyRDN}
    ::=  id-ase-chainedModifyASE

chainedAddEntry     ChainedAddEntry     ::= 6
chainedRemoveEntry  ChainedRemoveEntry  ::= 7
chainedModifyEntry  ChainedModifyEntry  ::= 8
chainedModifyRDN    ChainedModifyRDN    ::= 9
```

7.3 *Contextos de aplicación de la guía*

7.3.1 *Contexto de aplicación de acceso a la guía*

El **directoryAccessAC** autoriza al AUG a ganar acceso a las operaciones de los siguientes ESA: **readASE**, **searchASE**, **modifyASE**.

```
directoryAccessAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS
            {aCSE}
        BIND DirectoryBind
        UNBIND DirectoryUnbind
        REMOTE OPERATIONS {rOSE}
        INITIATOR CONSUMER OF {
            readASE,
            searchASE,
            modifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse,
            id-as-directoryAccessAS}
    ::=  id-ac-directoryAccessAC
```

7.3.2 *Contexto de aplicación de sistema de guía*

El **directorySystemAC** permite a los ASG comunicar con el fin de concatenar operaciones.

```
directorySystemAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS
            {aCSE}
        BIND DSABind
        UNBIND DSAUnbind
```

```
        REMOTE OPERATIONS {rOSE}
        OPERATIONS OF
            {chainedReadASE,
            chainedSearchASE,
            chainedModifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse,
            id-as-directorySystemAS}
    ::= id-ac-directorySystemAC
```

7.4 *Errores*

Correspondientemente a cada error abstracto definido en el servicio abstracto hay un valor de error que puede ser transportado por el protocolo. Se aplican las siguientes asignaciones:

| | | | |
|---|---|---|---|
| **abandoned** | **Abandoned** | ::= | **5** |
| **attributeError** | **AttributeError** | ::= | **1** |
| **nameError** | **NameError** | ::= | **2** |
| **referral** | **Referral** | ::= | **4** |
| **securityError** | **SecurityError** | ::= | **6** |
| **serviceError** | **ServiceError** | ::= | **3** |
| **updateError** | **UpdateError** | ::= | **8** |
| **dSAReferral** | **DSAReferral** | ::= | **9** |
| **abandonFailed** | **AbandonFailed** | ::= | **7** |

## 8 Correspondencia con los servicios utilizados

Este punto define la correspondencia del PAG y del PSG con los servicios utilizados.

### 8.1 *Correspondencia con ESCA*

Este punto define la correspondencia del servicio abstracto de vincular (**DirectoryBind** o **DSABind**) y el servicio abstracto de desvincular (**DirectoryUnbind** o **DSAUnbind**) con los servicios de ESCA. El ESCA se define en la Recomendación X.217.

#### 8.1.1 *Vincular-abstracto con A-ASOCIACION*

El servicio abstracto de vincular corresponde con el servicio A-ASOCIACION del ESCA. La utilización de los parámetros del servicio A-ASOCIACION va calificada por los siguientes puntos.

##### 8.1.1.1 *Modo*

Este parámetro lo suministrará el iniciador de la asociación en la primitiva petición de A-ASOCIACION, y tendrá el valor "modo normal".

##### 8.1.1.2 *Nombre de contexto de aplicación*

El iniciador de la asociación propondrá el contexto de aplicación **directoryAccessAC** o el **directorySystemAC.**

##### 8.1.1.3 *Información de usuario*

La correspondencia de la operación de vinculación del servicio abstracto de vinculación con los parámetros de información de usuario de la primitiva petición de A-ASOCIACION se define en la Recomendación X.219.

##### 8.1.1.4 *Lista de definiciones del contexto de presentación*

El iniciador de la asociación suministrará la lista de definiciones del contexto de presentación en la primitiva petición de A-ASOCIACION, que contendrá el sistema abstracto ESCA (**id-as-acse**) y, o bien la sintaxis abstracta PAG (**id-as-directory/AccessAS**) o la sintaxis abstracta ESC (**id-as-directorysystemAS**).

8.1.1.5 *Calidad de servicio*

Este parámetro lo suministrará el iniciador de la asociación en la primitiva petición de A-ASOCIACION, y el respondedor de la asociación en la primitiva respuesta de A-ASOCIACION. Los parámetros "control ampliado" y "transferencia de diálogo optimizada" se fijarán a "característica no requerida". Los parámetros restantes tomarán sus respectivos valores por defecto.

8.1.1.6 *Requisitos de sesión*

Este parámetro lo fijará el iniciador de la asociación en la primitiva petición de A-ASOCIACION, y el respondedor de la asociación en la primitiva respuesta de A-ASOCIACION. Este parámetro se fijará de modo que especifique las siguientes unidades funcionales:

a) Kernel;

b) Dúplex.

8.1.1.7 *Título de entidad de aplicación y dirección de presentación*

Estos parámetros los proporcionarán el iniciador y el respondedor de la asociación (el suministro del título de entidad de aplicación es optativo. Para un AUG que establezca una asociación para una petición inicial, estos parámetros se obtendrán a partir de informaciones mantenidas localmente.

Para que un AUG (o un ASG) que establezca una asociación con un ASG que se le ha dado como referencia, estos parámetros se obtienen a partir del valor de **AccessPoint** de una **ContinuationReference**. Para un ASG que establezca una asociación, este parámetro se obtiene a partir de su información de conocimiento, es decir, una referencia externa.

8.1.2 *Desvincular-abstracto con A-LIBERACION*

El servicio abstracto de desvincular corresponde con el servicio A-LIBERACION del ESCA. La utilización de los parámetros del servicio A-LIBERACION va calificada con el siguiente punto.

8.1.2.1 *Resultado*

Este parámetro tendrá el valor "afirmativo".

8.1.3 *Utilización de los servicios A-ABORTO y A-P-ABORTO*

El proceso de aplicación es el usuario de los servicios A-ABORTO y A-P-ABORTO en ESCA.

8.2 *Correspondencia con ESOD*

Los servicios ESA de la guía corresponden con los servicios OD-INVOCACION, OD-RESULTADO, OD-ERROR, OD-RECHAZO-U y OD-RECHAZO-P del ESOD. La correspondencia de la notación de sintaxis abstracta de los ESA de la guía con los servicios ESOD se define en la Recomendación X.219.

## 9 Conformidad

Este punto define los requisitos de la conformidad con esta Recomendación.

9.1 *Conformidad de los AUG*

Una implementación de AUG que anuncia conformidad con esta Recomendación deberá satisfacer los requisitos especificados en los § 9.1.1 a 9.1.3.

9.1.1 *Requisitos del enunciado de conformidad (o declaración de conformidad)*

Se enunciará lo siguiente:

a) las operaciones del contexto de aplicación de **directoryAccessAC** que el AUG es capaz de invocar y con relación a las cuales se anuncia la conformidad; y

b) el nivel o los niveles de seguridad con relación a los cuales se anuncia conformidad (ninguno, simple, fuerte).

9.1.2 *Requisitos estáticos*

Un AUG deberá:

a) tener la capacidad para soportar el contexto de aplicación directoryAccessAC, como se define por su sintaxis abstracta en el § 7.

9.1.3 *Requisitos dinámicos*

Un AUG deberá:

a) cumplir la correspondencia con los servicios utilizados definidos en el § 8.

9.2 *Conformidad de los ASG*

Una implementación de ASG que anuncia su conformidad con esta Recomendación deberá satisfacer los requisitos especificados en los § 9.2.1 a 9.2.3.

9.2.1 *Requisitos del enunciado de conformidad (o declaración de conformidad)*

Se enunciará lo siguiente:

a) Los contextos de aplicación con relación a los cuales se anuncia conformidad: el contexto de aplicación de acceso a la guía (**directoryAccessAC**), el contexto de aplicación de sistema de guía (**directorySystemAC**), o ambos.

   Si un ASG reúne condiciones tales que el conocimiento del mismo se ha divulgado y ha dado lugar a que referencias de conocimiento sobre dicho ASG estén contenidas en otros ASG, fuera del dominio de gestión de la guía (DGG), deberá anunciar su conformidad con el contexto de aplicación de sistema de guía.

   *Nota* - Un contexto de aplicación no se dividirá, salvo en las circunstancias aquí enunciadas: específicamente, no se puede anunciar conformidad con puertos u operaciones particulares.

b) Si el ASG puede o no actuar como ASG de primer nivel, como se define en la Recomendación X.518;

c) si se anuncia la conformidad con el contexto de aplicación de sistema de guía (**directorySystemAC**), deberá precisarse si se soporta o no el modo concatenado, definido en la Recomendación X.518;

d) el nivel o los niveles de seguridad con relación a los cuales se anuncia conformidad (ninguno, simple, fuerte);

e) los tipos de atributo seleccionados, definidos en la Recomendación X.520, y cualesquiera otros tipos de atributo con relación a los cuales se anuncie conformidad; y

f) las clases de objeto seleccionadas definidas en la Recomendación X.521 y cualesquiera otras clases de objeto con relación a las cuales se anuncie conformidad.

9.2.2 *Requisitos estáticos*

Un ASG deberá:

a) tener la capacidad para soportar los contextos de aplicación con relación a los cuales se anuncia conformidad, como se define por su sintaxis abstracta en el § 7;

b) tener la capacidad para soportar el marco de información definido por su sintaxis abstracta en la Recomendación X.501;

c) ser conforme con los requisitos de conocimiento mínimo definidos en la Recomendación X.518;

d) si se anuncia conformidad como ASG del primer nivel, deberá anunciarse la conformidad con los requisitos que deben cumplirse para soportar el contexto de raíz, definido en la Recomendación X.518;

e) tener la capacidad para soportar los tipos de atributo con relación a los cuales se anuncia conformidad como se define por su sintaxis abstracta; y

f) tener la capacidad para soportar las clases de objeto en relación a las cuales se anuncia conformidad, como se define por su sintaxis abstracta.

9.2.3 *Requisitos dinámicos*

Un ASG deberá:

a) cumplir la correspondencia con los servicios utilizados definidos en el § 8 de esta Recomendación;

b) ser conforme con los procedimientos para la operación distribuida de la guía, relacionados con referimientos, definidos en la Recomendación X.518;

c) si se anuncia conformidad con el contexto de aplicación **directoryAccessAC**, la conformidad deberá incluir los procedimientos de la Recomendación X.518 en cuanto se relacionen con el modo referimiento del PAG;

d) si se anuncia de conformidad con el contexto de aplicación **directorySystemAC**, la conformidad deberá incluir el modo de interacción por referimiento, definido en la Recomendación X.518.

e) si se anuncia conformidad con el modo concatenado de interacción, la conformidad deberá incluir el modo de interacción concatenado, definido en la Recomendación X.518.

*Nota* - Sólo en este caso es necesario que el ASG pueda invocar operaciones utilizando el contexto de aplicación **directorySystemAC**.

## ANEXO A

(a la Recomendación X.519)

## PAG en NSA.1

Este anexo forma parte integrante de la Recomendación.

Este anexo incluye todas las definiciones de tipo y valor NSA.1 contenidas en esta Recomendación en forma del módulo NSA.1, **DirectoryAccessProtocol**.

```
DirectoryAccessProtocol {joint-iso-ccitt ds(5) modules(1) dap(11)}
DEFINITIONS ::=
BEGIN

EXPORTS
    directoryAccessAC, readASE, searchASE, modifyASE;

IMPORTS
    abstractService
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

    APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, aCSE
    FROM    Remote-Operations-Notation-extension
            {joint-iso-ccitt remoteOperations(4) notation-extension(2)}

    id-ac-directoryAccessAC, id-ase-readASE, id-ase-searchASE,
    id-ase-modifyASE, id-as-directoryAccessAS, id-as-acse
    FROM    ProtocolObjectIdentifiers
            {joint-iso-ccitt  ds(5) modules(1)
            protocolObjectIdentifiers(4)}

    DirectoryBind, DirectoryUnbind, Read, Compare, Abandon, List,
    Search, AddEntry, RemoveEntry, ModifyEntry, ModifyRDN, Abandoned, AbandonFailed,
    AttributeError, NameError, Referral, SecurityError, ServiceError,
    UpdateError
    FROM    DirectoryAbstractService
            directoryAbstractService;

-- Contextos de aplicación --

directoryAccessAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS {aCSE}
        BIND DirectoryBind
```

```
        UNBIND DirectoryUnbind
        REMOTE OPERATIONS {rOSE}
            INITIATOR CONSUMER OF {readASE, searchASE, modifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse, id-as-directoryAccessAS}
    ::= id-ac-directoryAccessAC

-- Leer ESA (ASE) --

readASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES {read, compare, abandon}
    ::= id-ase-readASE

-- Buscar ESA (ASE) --

searchASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES {list, search}
    ::= id-ase-searchASE

-- Modificar ESA (ASE) --

modifyASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
            {addEntry, removeEntry,
            modifyEntry, modifyRDN}
    ::= id-ase-modifyASE

-- Operaciones distantes --

read          Read          ::= 1
compare       Compare       ::= 2
abandon       Abandon       ::= 3
list          List          ::= 4
search        Search        ::= 5
addEntry      AddEntry      ::= 6
removeEntry   RemoveEntry   ::= 7
modifyEntry   ModifyEntry   ::= 8
modifyRDN     ModifyRDN     ::= 9

-- Errores distantes --

attributeError   AttributeError      ::= 1
nameError        NameError           ::= 2
serviceError     ServiceError        ::= 3
referral         Referral            ::= 4
abandoned        Abandoned           ::= 5
securityError    SecurityError       ::= 6
abandonFailed    AbandonedFailed     ::= 7
updateError      UpdateError         ::= 8
END
```

ANEXO B

(a la Recomendación X.519)

**PSG en NSA.1**

Este anexo forma parte integrante de la Recomendación.

Este anexo incluye todas las definiciones de tipo y valor NSA.1 contenidas en esta Recomendación en forma del módulo NSA.1, **DirectorySystemProtocol.**

```
DirectorySystemProtocol {joint-iso-ccitt ds(5) modules(1) dsp(12)}
DEFINITIONS ::=
BEGIN

EXPORTS
    directorySystemAC, chainedReadASE, chainedSearchASE,
    chainedModifyASE;

IMPORTS
    distributedOperations, directoryAbstractService
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

    APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, aCSE
    FROM    Remote-Operations-Notation-extension
            {joint-iso-ccitt remoteOperations(4) notation-extension(2)}

    id-ac-directorySystemAC, id-ase-chainedReadASE,
    id-ase-chainedSearchASE, id-ase-chainedModifyASE,
    id-as-directorySystemAS, id-as-acse;
    FROM    ProtocolObjectIdentifiers
            {joint-iso-ccitt  ds(5) modules(1)
            protocolObjectIdentifiers(4)}

    Abandoned, AttributeError, AbandonFailed,
    NameError, DSAReferral, SecurityError, ServiceError, UpdateError
    FROM    DirectoryAbstractService directoryAbstractService

    DSABind, DSAUnbind,
    ChainedRead, ChainedCompare, ChainedAbandon,
    ChainedList, ChainedSearch,
    ChainedAddEntry, ChainedRemoveEntry, ChainedModifyEntry,
    ChainedModifyRDN, DSAReferral
    FROM    DistributedOperations
            distributedOperations;

-- Contextos de aplicación --

directorySystemAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS {aCSE}
        BIND DSABind
        UNBIND DSAUnbind
        REMOTE OPERATIONS {rOSE}
            OPERATIONS OF {
                chainedReadASE, chainedSearchASE, chainedModifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse, id-as-directorySystemAS}
    ::= {id-ac-directorySystemAC}

-- Concatenado-leer ESA (ASE) --

chainedReadASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS          {chainedRead, chainedCompare, chainedAbandon}
    ::= id-ase-chainedReadASE
```

```
-- Concatenado-buscar ESA (ASE) --

chainedSearchASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS          {chainedList, chainedSearch}
    ::= id-ase-chainedSearchASE

-- Concatenado modificar ESA (ASE) --

chainedModifyASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS
            {chainedAddEntry, chainedRemoveEntry,
            chainedModifyEntry, chainedModifyRDN}
    ::= id-ase-chainedModifyASE

-- Operaciones distantes --

chainedRead             ChainedRead             ::= 1
chainedCompare          ChainedCompare          ::= 2
chainedAbandon          ChainedAbandon          ::= 3
chainedlist             ChainedList             ::= 4
chainedSearch           ChainedSearch           ::= 5
chainedAddEntry         ChainedAddEntry         ::= 6
chainedRemoveEntry      ChainedRemoveEntry      ::= 7
chainedModifyEntry      ChainedModifyEntry      ::= 8
chainedModifyRDN        ChainedModifyRDN        ::= 9

-- Errores distantes --

attributeError      AttributeError      ::= 1
nameError           NameError           ::= 2
serviceError        ServiceError        ::= 3
abandoned           Abandoned           ::= 5
securityError       SecurityError       ::= 6
abandonFailed       AbandonFailed       ::= 7
updateError         UpdateError         ::= 8
dsaReferral         DSAReferral         ::= 9
END
```

## ANEXO C

(a la Recomendación X.519)

### Definición de referencia de los identificadores de objeto para protocolos

Este anexo forma parte de la Recomendación.

Este anexo incluye todos los identificadores de objeto NSA.1 asignados en esta Recomendación en forma de módulo NSA.1, **ProtocolObjetIdentifiers.**

```
ProtocolObjectIdentifiers {joint-iso-ccitt ds(5) modules(1) protocolObjectIdentifiers(4)}
DEFINITIONS ::=
BEGIN
```

```
EXPORTS
    id-ac-directoryAccessAC, id-ac-directorySystemAC, id-ase-readASE, id-ase-searchASE,
    id-ase-modifyASE, id-ase-chainedReadASE,
    id-ase-chainedSearchASE, id-ase-chainedModifyASE, id-as-acse,
    id-as-directoryAccessAS, id-as-directorySystemsAS;

IMPORTS
    id-ac, id-ase, id-as
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)};

-- Application Contexts --

id-ac-directoryAccessAC   OBJECT IDENTIFIER ::= {id-ac 1}

id-ac-directorySystemAC   OBJECT IDENTIFIER ::= {id-ac 2}

-- ASEs --

id-ase-readASE            OBJECT IDENTIFIER ::= {id-ase 1}

id-ase-searchASE          OBJECT IDENTIFIER ::= {id-ase 2}

id-ase-modifyASE          OBJECT IDENTIFIER ::= {id-ase 3}

id-ase-chainedReadASE     OBJECT IDENTIFIER ::= {id-ase 4}

id-ase-chainedSearchASE   OBJECT IDENTIFIER ::= {id-ase 5}

id-ase-chainedModifyASE   OBJECT IDENTIFIER ::= {id-ase 6}

-- ASs --

id-as-directoryAccessAS   OBJECT IDENTIFIER ::= {id-as 1}

id-as-directorySystemAS   OBJECT IDENTIFIER ::= {id-as 2}

id-as-acse                OBJECT IDENTIFIER ::=
                          {joint-iso-ccitt association-control(2) abstract-syntax(1) apdus(0)
                          version1(1)}

END
```

**Recomendación X.520**

## LA GUIA - TIPOS DE ATRIBUTO SELECCIONADOS [1)]

*(Melbourne, 1988)*

INDICE



---

1) La Recomendación X.520 y la norma ISO 9594-6, Information Processing Systems - Open Systems Interconnection - The Directory - Selected attribute types (Sistemas de procesamiento de información - Interconexión de sistemas abiertos La guía - Tipos de atributo seleccionados) se redactaron en estrecha colaboración y están técnicamente alineadas.

## 0 Introducción

0.1 Esta Recomendación, junto con las demás de la serie, ha sido elaborada para facilitar la interconexión de los sistemas de tratamiento de la información con objeto de ofrecer servicios de guía. Un conjunto de tales sistemas, además de la información de guía que contienen, puede considerarse como un todo integrado, llamado la *guía*. La información que contiene la guía, conocida colectivamente como base de información de la guía (BIG) suele utilizarse para facilitar la comunicación entre, con o sobre objetos tales como entidades de aplicación, personas, terminales y listas de distribución.

0.2 La guía desempeña un papel importante en la interconexión de sistemas abiertos (ISA) cuya finalidad es permitir, con un mínimo de acuerdo técnico aparte de las normas de interconexión en sí mismas, la interconexión de sistemas de tratamiento de la información:

- de diferentes fabricantes;
- sujetos a gestiones diferentes;
- de diferentes grados de complejidad; y
- de diferentes fechas de construcción.

0.3 En esta Recomendación se definen varios tipos de atributos que pueden resultar de utilidad para una serie de aplicaciones propias de la guía. Muchos de los atributos aquí definidos se utilizan en particular para la formación de nombres, especialmente para las clases de objetos definidas en la Recomendación X.521. En esta Recomendación se definen asimismo varias sintaxis de atributo normalizadas.

0.4 El anexo A, que forma parte de la presente Recomendación, proporciona la notación NSA.1 para el módulo completo donde se definen los atributos y la sintaxis de atributo.

0.5 El anexo B, que no forma parte de la presente Recomendación, proporciona un índice alfabético de los tipos de atributo para facilitar la referencia a los mismos.

## 1 Alcance y campo de aplicación

1.1 En esta Recomendación se definen varios tipos de atributo que pueden resultar de utilidad para una gama de aplicaciones de la guía.

1.2 Los tipos de atributo (y la sintaxis de atributo) se dividen en tres categorías, como se indica en los § 1.2.1 a 1.2.3.

1.2.1 Algunos tipos (sintaxis) de atributo se utilizan en una amplia variedad de aplicaciones o son comprendidos y/o utilizados por la propia guía.

*Nota* - Se recomienda utilizar un tipo (sintaxis) de atributo definido en este documento antes que crear otro nuevo, siempre que sea apropiado para la aplicación de que se trate.

1.2.2 Algunos tipos (sintaxis) de atributo están normalizados internacionalmente pero son específicos para una aplicación. Estos tipos se definen en las normas relativas a la aplicación correspondiente.

1.2.3 Cualquier autoridad administrativa puede definir sus propios tipos (sintaxis) de atributo para cualquier finalidad. Estos no están normalizados internacionalmente y pueden ponerse a disposición de terceros, aparte de la autoridad administrativa que los creó, sólo mediante acuerdo bilateral.

## 2 Referencias

ISO 3166 - Codes for the representation of names of countries

Recomendación X.121 - Plan de numeración internacional para redes públicas de datos

Recomendación X.208 - Especificación de la notación de sintaxis abstracta uno (NSA.1) (véase también ISO 8824)

Recomendación X.501 - La guía - Modelos (véase también ISO 9594-2)

Recomendación X.521 - La guía - Clases de objeto seleccionadas (véase también ISO 9594-7)

Recomendación E.123 - Notación para los números telefónicos nacionales e internacionales

## 3 Definiciones

En la presente Recomendación se utilizan las siguientes definiciones de la Recomendación X.501:

a) *tipo de atributo*;

b) *sintaxis de atributo*;

c) *clase de objeto.*

## 4 Notación

Los tipos de atributo y la sintaxis de atributo se definen en este documento mediante la utilización de una notación especial definida como macros NSA.1 en la Recomendación X.501. Hay dos de estas macros, **ATTRIBUTE** y **ATTRIBUTE-SYNTAX.**

Se utilizan dos identificadores de objeto genéricos (**attributeType** y **attributeSyntax**) para definir los identificadores de objeto asignados respectivamente a tipos de atributo y sintaxis de atributo. Sus definiciones figuran en el anexo B de la Recomendación X.501.

Los ejemplos del uso de los tipos de atributo se describen mediante una notación informal, donde los pares formados por el tipo y el valor del atributo se representan por un acrónimo del tipo de atributo, seguido de un signo igual ("="), seguido del valor del atributo en el ejemplo.

SECCION 1 - *Tipos de atributo seleccionados*

## 5 Definición de tipos de atributo seleccionados

En esta Recomendación se definen varios tipos de atributo que pueden resultar de utilidad para una gama de aplicaciones de la guía.

5.1 *Tipos de atributo de sistema*

Esos tipos de atributo están relacionados con la información sobre los objetos conocidos por la guía.

5.1.1 *Clase de objeto*

El tipo de atributo *clase de objeto*, que la guía conoce, está definido, excepto para la asignación de un identificador de objeto, en la Recomendación X.501.

**objectClass ObjectClass ::= {attributeType 0}**

5.1.2 *Nombre de objeto con alias*

Este tipo de atributo está definido, excepto para la asignación de un identificador de objeto, en la Recomendación X.501.

**aliasedObjectName AliasedObjectName ::= (attributeType 1)**

5.1.3 *Información de Conocimiento*

El tipo de atributo *información de conocimiento* contiene una descripción acumulada, legible para el ser humano, del conocimiento que posee un determinado ASG.

**knowledgeInformation ATTRIBUTE**
**WITH ATTRIBUTE-SYNTAX caseIgnoreStringSyntax**
**::= (attributeType 2)**

5.2 *Tipo de atributo de etiquetado*

Estos atributos conciernen a la información sobre objetos que ha sido explícitamente asociada con los objetos mediante un proceso de etiquetado.

5.2.1 *Nombre común*

El tipo de atributo *nombre común* determina el identificador de un objeto que ha sido denominado. Un nombre común no es un nombre de la guía; es un nombre (posiblemente ambiguo) por el que suele conocerse el objeto en algún ámbito limitado (por ejemplo, una organización) y que es conforme a los convenios de nominación del país o a las convenciones de la cultura a que está asociado.

Un valor de atributo para un nombre común es una cadena elegida por la persona u organización descrita por el nombre, o elegida por la organización responsable del objeto descrito por el nombre, para dispositivos y entidades de aplicación. Por ejemplo, un nombre típico de una persona en un país de habla inglesa se compone de un título personal (por ejemplo Mr, Ms, Dr, Professor, Sir, Lord), un primer nombre, uno o más nombres intermedios, un apellido, un calificativo generacional (eventualmente, por ejemplo Jr.) distinciones honoríficas y condecoraciones (eventualmente, por ejemplo QC).

Ejemplos:

NC: "Mr Robin Lachlan McLeod BSc (Hons) CEng MIEE"

NC: "Divisional Coordination Committee"

NC: "High Speed Modem"

Cualquier variante será asociada con el objeto denominado en tanto que valores de atributo separados y alternativos.

Deberán también admitirse otras variantes comunes, tales como la utilización de un segundo nombre intermedio como primer nombre preferido, o la utilización de "Bill" en lugar de "William", etc.

**commonName ATTRIBUTE**
**WITH ATTRIBUTE-SYNTAX**
**caseIgnoreStringSyntax**
**(SIZE(1..ub-common-name))**
**::= {attributeType 3}**

5.2.2 *Apellido*

El tipo de atributo *apellido* especifica la construcción lingüística que generalmente se hereda de los padres o se asume por casamiento, y por la que suele conocerse a la persona.

Un valor de atributo para apellido es una cadena, por ejemplo, "McLeod".

```
surname ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-surname))
    ::= {attributeType 4}
```

5.2.3 *Número de serie*

El tipo de atributo *número de serie* especifica un identificador, el número de serie de un dispositivo.

Un valor de atributo para número de serie es una cadena imprimible.

```
serialNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        printableStringSyntax
            (SIZE(1..ub-serial-number))
    ::= {attributeType 5}
```

5.3 *Tipos de atributo geográficos*

Estos tipos de atributo se refieren a posiciones geográficas o regiones con las que están asociados los objetos.

5.3.1 *Nombre de país*

El tipo de atributo *nombre de país* especifica un país. Cuando se usa como componente de un nombre de guía, expresa el país en que está ubicado físicamente el objeto denominado o con el que se encuentra asociado de alguna otra manera importante.

Un valor de atributo para nombre de país es una cadena elegida de ISO 3166.

```
countryName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PrintableString (SIZE (2)) - IS 3166 codes only
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {attributeType 6}
```

La regla de concordancia para este tipo es la misma que se utiliza para **caseIgnoreStringSyntax.**

5.3.2 *Nombre de localidad*

El *tipo de atributo* nombre de localidad especifica una localidad. Cuando se usa como componente de un nombre de guía, expresa una zona geográfica o localidad en la que está ubicado físicamente el objeto denominado o con la cual se encuentra asociado de alguna otra manera importante.

Un valor de atributo para el nombre de localidad es una cadena: por ejemplo, L = "Edinburgo".

```
localityName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-locality-name))
    ::= {attributeType 7}
```

5.3.3 *Nombre de estado o provincia*

El tipo de atributo *nombre de estado o provincia* especifica un estado o una provincia. Cuando se usa como componente de un nombre de guía, determina una subdivisión geográfica en la que está ubicado físicamente el objeto denominado, o con la que éste se encuentra asociado de alguna otra manera importante.

Un valor de atributo para nombre de estado o provincia es una cadena, por ejemplo, S = "Ohio".

```
stateOrProvinceName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-state-name))
    ::= {attributeType 8}
```

5.3.4 *Dirección-calle*

El tipo de atributo *dirección-calle* especifica un lugar para la distribución y la entrega material a una dirección postal, o sea nombre de la calle, lugar, avenida, y el número del edificio. Cuando se usa como componente de un nombre de guía, expresa la dirección de calle en que está ubicado el objeto denominado, o con la que éste se encuentra asociado de alguna otra manera importante.

Un valor de atributo para dirección-calle es una cadena, por ejemplo: "Arnulfstraße 60".

```
streetAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-street-address))
    ::= {attributeType 9}
```

5.4 *Tipos de atributo organizacionales*

Estos tipos de atributo se refieren a las organizaciones y se pueden utilizar para describir objetos en base a las organizaciones con las que están asociados.

5.4.1 *Nombre de organización*

El tipo de atributo *nombre de organización* especifica una organización. Cuando se usa como componente de nombre de guía, expresa una organización a la que pertenece el objeto denominado.

Un valor de atributo para nombre de organización es una cadena escogida por la organización (por ejemplo 0 = "Scottish Telecommunications plc"). Las variantes eventuales deberán asociarse con la organización denominada como valores de atributo separados y alternativos.

```
organizationName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-organization-name))
    ::= {attributeType 10}
```

5.4.2 *Nombre de unidad organizacional*

El tipo de atributo *nombre de unidad organizacional* especifica una unidad orgánica. Cuando se usa como componente de un nombre de guía, expresa una unidad orgánica a la que pertenece el objeto denominado.

Se entenderá que la unidad organizacional designada formará parte de una organización designada por un atributo OrganizationName.

De ahí que si en un nombre se utiliza un atributo nombre de unidad organizacional, este deberá estar asociado con un atributo OrganizationName.

Un valor de atributo para el nombre de unidad organizacional es una cadena escogida por la organización de la que forma parte (por ejemplo UO = "División Tecnología"). Obsérvese que la abreviatura de uso corriente "DT" sería un valor de atributo separado y alternativo.

Ejemplos:

0 = "Scottel", UO = "DT"

```
organizationalUnitName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-organizational-unit-name))
    ::= {attributeType 11}
```

5.4.3 *Título*

El tipo de atributo *título* especifica el cargo o función designado del objeto dentro de una organización.

Un valor de atributo para título es una cadena.

Ejemplo:

T = "Gestor, Aplicaciones Distribuidas"

```
title ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-title))
    ::= {attributeType 12}
```

5.5 *Tipos de atributo explicativos*

Estos tipos de atributo se refieren a explicaciones (por ejemplo, en un lenguaje ordinario) sobre algo relativo a un objeto.

5.5.1 *Descripción*

El tipo de atributo *descripción* contiene texto que describe el uso del objeto asociado.

Por ejemplo, el objeto "interés normativo" puede llevar la descripción asociada "lista de distribución para el intercambio de informaciones acerca de la elaboración de normas internas de la compañía".

Un valor de atributo para descripción es una cadena.

```
description ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-description))
    ::= {attributeType 13}
```

5.5.2 *Directriz de búsqueda*

El tipo de atributo *directriz de búsqueda* (Search Guide) especifica la información sobre los criterios de búsqueda propuestos que se podrán incluir en algunos asientos que se espera constituirán un objeto base útil para la operación de búsqueda por ejemplo, un país u organización.

Los criterios de búsqueda consisten en un identificador facultativo para el tipo de objeto buscado, y en combinaciones de tipos de atributo y operadores lógicos que se utilizarán para construir un filtro. Para cada criterio de búsqueda es posible especificar el nivel de concordancia, por ejemplo concordancia aproximada.

El atributo directriz de búsqueda puede repetirse para reflejar diversos tipos de petición, por ejemplo búsqueda de una Persona Residencial o una Persona Organizacional, que se pueden cumplir a partir del objeto base dado donde se lee la directriz de búsqueda.

```
searchGuide ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        Guide
     ::= {attributeType 14}

Guide ::= SET {
    objectClass     [0]  OBJECT-CLASS OPTIONAL,
    criteria        [1]  Criteria}

Criteria ::=
    CHOICE{
        Type        [0]  CriteriaItem,
        and         [1]  SET OF Criteria,
        or          [2]  SET OF Criteria,
        not         [3]  Criteria}

CriteriaItem
    CHOICE {
        equality            [0]  AttributeType,
        substrings          [1]  AttributeType,
        greaterOrEqual      [2]  AttributeType,
        lessOrEqual         [3]  AttributeType,
        approximateMatch    [4]  AttributeType}
```

Ejemplo: A continuación se presenta un valor potencial del atributo directriz de búsqueda que podría almacenarse en asientos de clase de objeto localidad para indicar cómo podrían encontrarse los asientos de la clase de objeto persona residencial.

```
residential-person-guide    Guide ::= {
        objectClass  residentialPerson,
        criteria and {
                type substrings commonName,
                type substrings streetAddress }}
```

A partir de este valor de directriz (Guide), la construcción de un filtro es muy sencilla.

El paso (1) produce el valor de filtro intermedio:

```
intermediate-filter          Filter      ::=    and {
        item substrings {
                type commonName,
                strings {any T61String "Dubois" }} - valor suministrado para nombre común

        item substrings {
                type streetAddress,
                strings {any T61String "Hugo" }}}  - valor suministrado para  dirección-calle
```

El paso (2) produce un filtro para concordar las entradas Persona Residencial.

```
residential-person-filter          Filter      ::= {
        and {
                item equality {
                        objectClass,
                        OBJECT-CLASS residentialPerson },
                intermediate-filter  }}
```

5.5.3 *Categoría profesional*

El atributo *categoría profesional* especifica información sobre la ocupación de algunos objetos usuales tales como personas. Por ejemplo, este atributo permite interrogar a la guía acerca de las personas que ejercen la misma profesión.

```
businessCategory ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                caseIgnoreStringSyntax
                        (SIZE(1..ub-business-category))
        ::= {attributeType 15}
```

5.6 *Tipos de atributo de direccionamiento postal*

Estos tipos de atributo se refieren a la información requerida para la entrega postal física a un objeto.

5.6.1 *Dirección postal*

El tipo de atributo *dirección postal* especifica la información de dirección necesaria para que la autoridad postal efectúe la entrega física de mensajes postales al objeto denominado.

Un valor de atributo para dirección postal estará normalmente compuesto por atributos seleccionados de la versión 1 de la dirección O/D postal no formateada del SMT conforme a la Recomendación F.401, y se limitará a 6 líneas de 30 caracteres cada una, incluido el nombre postal de país. En general, la información contenida en ese tipo de dirección puede incluir un nombre de destinatario, dirección-calle, ciudad, estado o provincia, código postal y quizás un número de apartado postal según las exigencias específicas del objeto denominado.

```
postalAddress ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX PostalAddress
        MATCHES FOR EQUALITY
        ::= {attributeType 16}

PostalAddress ::= SEQUENCE SIZE(1..ub-postal-line) OF CHOICE {
        T61String (SIZE(1..ub-postal-string)),
        PrintableString (SIZE(1..ub-postal-string))}
```

La regla de concordancia para los valores de este tipo es la misma que para **caseIgnoreListSyntax** (sintaxis de listas sin atender a mayúsculas/minúsculas).

5.6.2 *Código postal*

El tipo de atributo *código postal* especifica el código postal del objeto denominado. Cuando este valor de atributo está presente, formará parte de la dirección postal del objeto.

Un valor de atributo para código postal es una cadena.

```
postalCode ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-postal-code))
    ::= {attributeType 17}
```

5.6.3 *Apartado de correos*

El tipo de atributo *apartado de correos* especifica el apartado de correos a través del cual se operará sobre el objeto la entrega postal física. Cuando está presente, el valor de atributo es parte de la dirección postal del objeto.

```
postOfficeBox ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-post-office-box))
    ::= {attributeType 18}
```

5.6.4 *Nombre de oficina de entrega física*

El tipo de atributo *nombre de oficina de entrega física* especifica el nombre de la ciudad, pueblo, etc., donde está ubicada una oficina de entrega física.

Un valor de atributo de nombre de oficina de entrega física es una cadena.

```
physicalDeliveryOfficeName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-physical-office-name))
    ::= {attributeType 19}
```

5.7 *Tipos de atributo de direccionamiento de telecomunicaciones*

Estos tipos de atributo suministran información de direccionamiento necesaria para comunicar con el objeto mediante la telecomunicación.

5.7.1 *Número de teléfono*

El tipo de atributo *número de teléfono* especifica un número de teléfono asociado con un objeto.

Un valor de atributo para número de teléfono es una cadena conforme al formato internacionalmente convenido para indicar los números de teléfono internacionales y que figura en la Recomendación E.123 (por ejemplo, "+44 582 10101").

```
telephoneNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        telephoneNumberSyntax
    ::= {attributeType 20}
```

5.7.2 *Número télex*

El tipo de atributo *número télex* especifica el número téex, el indicativo de país, y el distintivo de un terminal télex asociado con un objeto.

```
telexNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX TelexNumber
    ::= {attributeType 21}
```

```
TelexNumber ::= SEQUENCE {
    telexNumber      PrintableString,
        (SIZE(1..ub-telex-number)),
    countryCode      PrintableString,
        (SIZE(1..ub-country-code)),
    answerback PrintableString,
    (SIZE(1..ub-answerback)) }
```

5.7.3 *Identificador de terminal teletex*

El tipo de atributo *identificador de terminal teletex* especifica el identificador de terminal teletex (y facultativamente, parámetros) para un terminal teletex asociado con un objeto.

Un valor de atributo para identificador de terminal teletex es una cadena que cumple los requisitos de la Recomendación F.200, y un conjunto opcional cuyos componentes son conformes a la Recomendación T.62.

```
teletexTerminalIdentifier ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        TeletexTerminalIdentifier
    ::= {attributeType 22}

TeletexTerminalIdentifier ::= SEQUENCE {
    teletexTerminal PrintableString
        (SIZE(1..ub-teletex-terminal-id)),
    parameters TeletexNonBasicParameters
        OPTIONAL}
```

5.7.4 *Número de teléfono de facsímil*

El tipo de atributo *número de teléfono de facsímil* especifica un número de teléfono de un terminal facsímil (y facultativamente sus parámetros) asociado con un objeto.

Un valor de atributo para el número de teléfono facsímil es una cadena conforme al formato internacionalmente convenido para la indicación de los números de teléfono internacionales de la Recomendación E.123 (por ejemplo "+81 3347 7418"), y una cadena de bits facultativa (formateada conforme a la Recomendación T.30).

```
facsimileTelephoneNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        Facsimile TelephoneNumber
    ::= {attributeType 23}

FacsimileTelephoneNumber ::= SEQUENCE{
    telephoneNumber PrintableString
        (SIZE(1..ub-telephone-number)),
    parameters G3FacsimileNonBasicParameters
        OPTIONAL}
```

5.7.5 *Dirección X.121*

El tipo de atributo dirección X.121 especifica una dirección definida por la Recomendación X.121, asociada con un objeto.

```
x121Address ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        NumericString
            (SIZE(1..ub-x121-address))
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeType 24}
```

Las reglas de concordancia para los valores de este tipo son iguales a las de **nunericStringSyntax**.

5.7.6 *Número internacional RDSI*

El tipo de atributo *número internacional RDSI* especifica una dirección RDSI asociada con un objeto.

Un valor de atributo para el número internacional RDSI es una cadena conforme al formato internacionalmente convenido para las direcciones RDSI según la Recomendación E.164.

```
internationalISDNNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        NumericString
            (SIZE(1..ub-isdn-address))
    ::= {attributeType 25}
```

La regla de concordancia para los valores de este tipo es igual a la de **numericStringSyntax.**

5.7.7 *Dirección registrada*

El tipo de atributo *dirección registrada* especifica un nemónico de una dirección asociada con un objeto ubicado en un lugar determinado de una ciudad. El nemónico se registra en el país al que pertenece la ciudad y se utiliza para el suministro del servicio público de telegramas (según la Recomendación F.1).

```
registeredAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX PostalAddress
    ::= {attributeType 26}
```

5.7.8 *Indicador de destino*

El tipo de atributo *indicador de destino* especifica (conforme a las Recomendaciones F.1 y F.3) el país y la ciudad asociados con el objeto (el destinatario), necesarios para suministrar el servicio público de telegramas.

Un valor de atributo para indicador de destino es una cadena.

```
destinationIndicator ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PrintableString
            SIZE(1..ub-destination-indicator))
    - alphabetical characters only
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeType 27}
```

Los valores de concordancia de este tipo son iguales a los de **caseIgnoreStringSyntax** (sintaxis de cadena sin atender a mayúsculas/minúsculas).

5.8 *Tipos de atributo de preferencia*

Estos tipos de atributo se refieren a las preferencias de un objeto.

5.8.1 *Método de entrega preferido*

El tipo de atributo *método de entrega preferido* especifica el orden de prioridad del objeto para el método que se utilizará para comunicarse con él.

```
preferredDeliveryMethod ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        SEQUENCE OF INTEGER {
            any-delivery-method     (0),
            mhs-delivery            (1),
            physical-delivery       (2),
            telex-delivery          (3),
            teletex-delivery        (4),
            g3-facsimile-delivery   (5),
            g4-facsimile-delivery   (6),
            ia5-terminal-delivery   (7),
            videotex-delivery       (8),
            telephone-delivery      (9)}
    SINGLE VALUE
    ::= {attributeType 28}
```

## 5.9 *Tipos de atributos de aplicación ISA*

Estos tipos de atributos tratan de la información relacionada con los objetos en la capa de aplicación de la ISA.

### 5.9.1 *Dirección de presentación*

El tipo de atributo *dirección de presentación* especifica la dirección de presentación asociada con un objeto que representa una entidad de aplicación de ISA.

Un valor de atributo para dirección de presentación es una dirección de presentación como se define en la Recomendación X.200.

```
presentationAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PresentationAddress
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {attributeType 29}

PresentationAddress ::=  SEQUENCE
    pSelector       [0]  OCTET STRING OPTIONAL,
    sSelector       [1]  OCTET STRING OPTIONAL,
    tSelector       [2]  OCTET STRING OPTIONAL,
    nAddresses      [3]  SET SIZE (I..MAX) OF OCTEC STRING}
```

La regla de concordancia para los valores de este tipo establece que una dirección de presentación concuerda con una almacenada únicamente si los selectores son iguales y las n direcciones presentadas son un subconjunto de las almacenadas.

### 5.9.2 *Contexto de aplicación soportado*

El tipo de atributo *contexto de aplicación soportado* especifica el identificador de objeto de un contexto de aplicación que es soportado por el objeto (una entidad de aplicación ISA).

```
supportedApplicationContext ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        objectIdentifierSyntax
    ::= {attributeType 30}
```

## 5.10 *Tipos de atributo relacionales*

Estos tipos de atributo se refieren a información concerniente a los objetos que están relacionados de cierta manera con un determinado objeto.

### 5.10.1 *Miembro*

El tipo de atributo *miembro* especifica un grupo de nombres asociados con el objeto.

Un valor de atributo para miembro es un nombre distinguido.

```
member ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::= {attributeType 31}
```

### 5.10.2 *Propietario*

El tipo de atributo *propietario* (Owner) especifica el nombre de algún objeto que tiene alguna responsabilidad sobre el objeto asociado.

Un valor de atributo para propietario es un nombre distinguido (que podría representar a un grupo de nombres) y puede repetirse (recurrencia).

```
owner ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::= {attributeType 32}
```

5.10.3 *Ocupante de rol*

El tipo de atributo *ocupante de rol* (Role Occupant) especifica el nombre de un objeto que desempeña un rol (papel) organizacional.

Un valor de atributo para ocupante de rol es un nombre distinguido.

**roleOccupant ATTRIBUTE**
**WITH ATTRIBUTE-SYNTAX**
**distinguishedNameSyntax**
**::= {attributeType 33}**

5.10.4 *Véase también*

El tipo de atributo *véase también* (See Also) especifica nombres de otros objetos de guía que pueden constituir (en algún sentido) otros aspectos del mismo objeto del mundo real.

Un valor de atributo para véase también es un nombre distinguido.

**seeAlso ATTRIBUTE**
**WITH ATTRIBUTE-SYNTAX**
**distinguishedNameSyntax**
**::= {attributeType 34}**

5.11 *Tipos de atributos de seguridad*

Estos tipos de atributo se refieren a la seguridad o a los privilegios de seguridad de un objeto. Estos tipos de atributo se especifican, excepto la asignación de un identificador de objeto, en la Recomendación X.509.

5.11.1 *Contraseña de usuario*

**userPassword UserPassword**
**::= {attributeType 35}**

5.11.2 *Certificado de usuario*

**userCertificate UserCertificate**
**::= {attributeType 36}**

5.11.3 *Certificado CA*

**cACertificate CACertificate**
**::= {attributeType 37}**

5.11.4 *Lista de revocaciones de autoridad*

**authorityRevocationList AuthorityRevocationList**
**::= {attributeType 38}**

5.11.5 *Lista de revocaciones de certificado*

**certificateRevocationList CertificateRevocationList**
**::= {attributeType 39}**

5.11.6 *Par de certificados recíprocos*

**crossCertificatePair CrossCertificatePair**
**::= {attributeType 40}**

SECCION 2 - *Sintaxis de atributo*

## 6 Definición de sintaxis de atributo

6.1 *Sintaxis de atributo utilizadas por la guía*

6.1.1 *Indefinido*

La sintaxis de atributo *indefinido* está destinada a atributos cuyos valores no se espera que sean comparados por la guía.

Especificar esta sintaxis de atributo para un atributo equivale a especificar el tipo de dato **ANY** y ninguna regla de concordancia en la macro **ATTRIBUTE**, para el atributo.

```
undefined ATTRIBUTE-SYNTAX
    ANY
    ::= {attributeSyntax 0}
```

6.1.2 *Nombre distinguido*

La sintaxis de atributo *nombre distinguido* está destinada a los atributos cuyos valores son nombres distinguidos. Está definida, con excepción de la asignación de un identificador de objeto, en la Recomendación X.501.

```
distinguishedNameSyntax DistinguishedNameSyntax
    ::= {attributeSyntax 1}
```

6.1.3 *Identificador de objeto*

La sintaxis de atributo *identificador de objeto* está destinada a atributos cuyos valores son identificadores de objeto. Está definida, con excepción de la asignación de un identificador de objeto, en la Recomendación X.501.

```
objectIdentifierSyntax ObjectIdentifierSyntax
    ::= {attributeSyntax 2}
```

6.2 *Sintaxis de atributo de cadena*

En las sintaxis especificadas en los § 6.2.1 a 6.2.4 los siguientes espacios se consideran no significativos:

- espacios iniciales (o sea, los que preceden al primer carácter imprimible);
- espacios finales (o sea, los que siguen al último carácter imprimible);
- espacios internos consecutivos en exceso de uno (se considera que múltiples espacios consecutivos equivalen a un solo carácter de espacio).

Los atributos conformes a estas sintaxis se almacenarán para la concordancia en una forma que omita aquellos espacios que no son significativos según estas reglas.

6.2.1 *Cadena exacta en posición mayúsculas/minúsculas*

La sintaxis de atributo *cadena exacta en posición mayúsculas/minúsculas* está destinada a los atributos cuyos valores son cadenas (ya sea cadenas T.61 o cadenas imprimibles), en las cuales el hecho de estar escritas en (la posición) mayúsculas o en (la posición) minúsculas es significativo a los efectos de una comparación (por ejemplo, "Dundee" y "DUNDEE" no concuerdan).

```
caseExactStringSyntax ATTRIBUTE-SYNTAX
    CHOICE {T61String, PrintableString}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 3}
```

Para que dos cadenas que poseen esta sintaxis concuerden por igualdad deben tener la misma longitud y los caracteres correspondientes deben ser idénticos. Una cadena imprimible se puede comparar con una cadena T.61. Cuando los caracteres correspondientes pertenecen ambos al juego de caracteres de cadena imprimible, la comparación se efectúa normalmente. En cambio, si el carácter en la cadena T.61 no pertenece al juego de caracteres de cadena imprimible, no hay concordancia.

6.2.2 *Cadena ignorar posición mayúsculas/minúsculas*

La sintaxis de atributo *cadena ignorar posición mayúsculas/minúsculas* está destinada a los atributos cuyos valores son cadenas (ya sean cadenas T.61 o cadenas imprimibles), pero en las que el hecho de estar escritas en mayúsculas o en minúsculas no tiene significado alguno a los efectos de la comparación (por ejemplo, "Dundee" y "DUNDEE" concuerdan).

**caseIgnoreStringSyntax ATTRIBUTE-SYNTAX**
**CHOICE {T61String, PrintableString}**
**MATCHES FOR EQUALITY SUBSTRINGS**
**::= {attributeSyntax 4}**

Las reglas para la concordancia son idénticas a las de la sintaxis de atributo cadena exacta en posición mayúsculas/minúsculas, salvo que los caracteres que sólo difieren en la posición mayúsculas/minúsculas se consideran idénticos.

6.2.3 *Cadena imprimible*

La sintaxis de atributo *cadena imprimible* está destinada a los atributos cuyos valores son cadenas imprimibles.

**printableStringSyntax ATTRIBUTE-SYNTAX**
**PrintableString**
**MATCHES FOR EQUALITY SUBSTRINGS**
**::= {attributeSyntax 5}**

Las reglas para establecer la concordancia son idénticas a las aplicables a cadena exacta en posición mayúsculas/minúsculas.

6.2.4 *Cadena numérica*

La sintaxis de atributo *cadena numérica* está destinada a atributos cuyos valores son cadenas numéricas.

**numericStringSyntax ATTRIBUTE-SYNTAX**
**NumericString**
**MATCHES FOR EQUALITY SUBSTRINGS**
**::= {attributeSyntax 6}**

Las reglas de concordancia son idénticas a las de sintaxis de atributo cadena exacta en posición mayúsculas/minúsculas, salvo que todos los caracteres de espacio son saltados al efectuar la comparación.

6.2.5 *Lista ignorar posición mayúsculas/minúsculas*

La sintaxis de atributo *lista ignorar posición mayúsculas/minúsculas* está destinada a los atributos cuyos valores son secuencias de cadenas (ya sea cadenas T.61 o cadenas imprimibles), pero donde no reviste importancia a los efectos de la comparación el hecho de que estén escritas en mayúsculas o en minúsculas.

**caseIgnoreListSyntax ATTRIBUTE-SYNTAX**
**SEQUENCE OF**
**CHOICE {T61String, PrintableString}**
**MATCHES FOR EQUALITY SUBSTRINGS**
**::= {attributeSyntax 7}**

Habrá concordancia por igualdad entre dos listas ignorar posición mayúsculas/minúsculas únicamente si cada una tiene el mismo número de cadenas y las cadenas correspondientes concuerdan. Esta última concordancia es como para cadena ignorar posición mayúsculas/minúsculas (§ 6.1.3).

6.3 *Sintaxis de atributo diversas*

6.3.1 *Booleano*

La sintaxis de atributo *booleano* está destinada a los atributos cuyos valores son Booleanos (es decir, representan verdadero o falso).

**booleanSyntax ATTRIBUTE-SYNTAX**
**BOOLEAN**
**MATCHES FOR EQUALITY**
**::= {attributeSyntax 8}**

Dos valores de atributo de esta sintaxis concuerdan por igualdad si ambos son verdaderos o ambos son falsos.

6.3.2 *Entero*

La sintaxis de atributo *entero* está destinada a los atributos cuyos valores son enteros.

**integerSyntax ATTRIBUTE-SYNTAX**
**INTEGER**
**MATCHES FOR EQUALITY ORDERING**
**::= {attributeSyntax 9}**

Dos valores de atributo de esta sintaxis concuerdan por igualdad si los enteros son los mismos. Se aplican las reglas de ordenamiento de enteros.

6.3.3 *Cadena de octetos*

La sintaxis de atributo *cadena de octetos* está destinada a los atributos cuyos valores son cadenas de octetos.

**octetStringSyntax ATTRIBUTE-SYNTAX**
**OCTET STRING**
**MATCHES FOR EQUALITY SUBSTRINGS ORDERING**
**::= {attributeSyntax 10}**

Para que dos cadenas que tienen esta sintaxis de atributo concuerden por igualdad, deben tener la misma longitud y los octetos correspondientes deben ser idénticos. El ordenamiento está determinado por la relación de ordenamiento que se da entre los primeros octetos que difieren al comparar las cadenas desde el comienzo.

6.3.4 *Tiempo UTC*

La sintaxis de atributo *tiempo UTC* está destinada a los atributos cuyos valores representan tiempo absoluto.

**uTCTimeSyntax ATTRIBUTE-SYNTAX**
**UTCTime**
**MATCHES FOR EQUALITY ORDERING**
**::= {attributeSyntax 11}**

Dos valores de atributo de esta sintaxis concuerdan por igualdad si representan el mismo tiempo (u hora). Un tiempo (u hora) anterior se considera "menos" que un tiempo (u hora) posterior.

6.3.5 *Número de teléfono*

La sintaxis de atributo *número de teléfono* está destinada a los atributos cuyos valores son números de teléfono.

**telephoneNumberSyntax ATTRIBUTE-SYNTAX**
**PrintableString**
**(SIZE(1..ub-telephone-number))**
**MATCHES FOR EQUALITY SUBSTRINGS**
**::= {attributeSyntax 12}**

Las reglas para la concordancia son idénticas a las de la sintaxis de atributo posición exacta mayúsculas/minúsculas, salvo que todos los caracteres de espacio y "-" son saltados al efectuar la comparación.

## ANEXO A

(a la Recomendación X.520)

**Tipos de atributo seleccionados en NSA.1**

Este anexo forma parte integrante de la Recomendación.

Este anexo comprende todas las definiciones de tipo y valor NSA.1 contenidas en la presente Recomendación, en forma del módulo NSA.1, **SelectedAttributeTypes**.

```
SelectedAttributeTypes      {joint-iso-ccitt ds(5) modules(1)
                             selectedAttributeTypes(5)}
DEFINITIONS ::=
BEGIN

-- exporta todo --

IMPORTS
    informationFramework, authenticationFramework, attributeType,
    upperBounds
        FROM    UsefulDefinitions    {joint-ISO-CCITT ds(5) modules(1)
                                      usefulDefinitions(0) },
    ATTRIBUTE, ATTRIBUTE-SYNTAX, AttributeType, OBJECT-CLASS,
    ObjectClass, AliasedObjectName,
    DistinguishedNameSyntax, ObjectIdentifierSyntax
        FROM    InformationFramework informationFramework
    G3FacsimileNonBasicParameters,
    TeletexNonBasicParameters
        FROM    MTSAbstractService  {joint-ISO-CCITT mhs-motis(6)
                                     mts(3) modules(0) mts-abstract-service(1)}
    UserCertificate, CACertificate, CrossCertificatePair,
      CertificateRevocationList,
    AuthorityRevocationList, UserPassword
        FROM    AuthenticationFramework, authenticationFramework
        ub-answerback,
        ub-common-name, ub-surname, ub-serial-number,
        ub-locality-name, ub-state-name,
        ub-street-address, ub-organization-name,
        ub-organizational-unit-name, ub-title,
        ub-description, ub-business-category, ub-postal-line,
        ub-postal-string, ub-postal-code, ub-post-office-box,
        ub-physical-office-name, ub-telex-number,
        ub-country-code, ub-teletex-terminal-id,
        ub-telephone-number, ub-x121-address,
        ub-international-isdn-number, ub-destination-indicator,
        ub-user-password
            FROM    UpperBounds upperBounds;

-- tipo de atributo --

objectClass    ObjectClass ::= {attributeType 0}

aliasedObjectName    AliasedObjectName ::= {attributeType 1}

knowledgeInformation ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX caseIgnoreStringSyntax
    ::= {attributeType 2}

commonName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-common-name))
    ::= {attributeType 3}

surname ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-surname))
    ::= {attributeType 4}

serialNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        printableStringSyntax
            (SIZE(1..ub-serial-number))
    ::= {attributeType 5}
```

```
countryName ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                PrintableString (SIZE(2)) -- códigos IS 3166 solamente
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= {attributeType 6}

localityName ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                caseIgnoreStringSyntax
                        (SIZE(1..ub-locality-name))
        ::= {attributeType 7}

stateOrProvinceName ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                caseIgnoreStringSyntax
                        (SIZE(1..ub-state-name))
        ::= {attributeType 8}

streetAddress ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                caseIgnoreStringSyntax
                        (SIZE(1..ub-street-address))
        ::= {attributeType 9}

organizationName ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                caseIgnoreStringSyntax
                        (SIZE(1..ub-organization-Name))
        ::= {attributeType 10}

organizationalUnitName ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                caseIgnoreStringSyntax
                        (SIZE(1..ub-organizational-unit-name))
        ::= {attributeType 11}

title ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                caseIgnoreStringSyntax
                        (SIZE(1..ub-title))
        ::= {attributeType 12}

description ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                caseIgnoreStringSyntax
                        (SIZE(1..ub-description))
        ::= {attributeType 13}

searchGuide ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                Criteria
        ::= {attributeType 14}

Guide ::= SET {
        objectClass [0] OBJECT-CLASS OPTIONAL,
                criteria [1] Criteria }

Criteria ::=
        CHOICE {
                type [0] CriteriaItem,
                and  [1] SET OF Criteria
                or   [2] SET OF Criteria
                not  [3] Criteria}

CriteriaItem ::=
        CHOICE {
                equality          [0] AttributeType
                substrings        [1] AttributeType
                greaterOrEqual    [2] AttributeType
                lessOrEqual       [3] AttributeType
                approximateMatch  [4] AttributeType
```

```
businessCategory ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-business-category))
    ::= {attributeType 15}

postalAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX PostalAddress
    MATCHES FOR EQUALITY
    ::= {attributeType 16}

PostalAddress ::= SEQUENCE SIZE(1..ub-postal-line) OF
    CHOICE {
        T61String (SIZE(1..ub-postal-string)),
        PrintableString (SIZE(1..ub-postal-string))}

postalCode ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-postal-code))
    ::= {attributeType 17}

postOfficeBox ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-post-office-box))
    ::= {attributeType 18}

physicalDeliveryOfficeName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-physical-office-name))
    ::= {attributeType 19}

telephoneNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        telephoneNumberSyntax
    ::= {attributeType 20}

telexNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX TelexNumber
    ::= {attributeType 21}

TelexNumber ::= SEQUENCE {
    telexNumber         PrintableString
        (SIZE(1..ub-telex-number)),
    countryCode         PrintableString,
        (SIZE(1..ub-country-code)),
    answerback PrintableString
        (SIZE(1..ub-answerback))}

teletexTerminalIdentifier ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        TeletexTerminalIdentifier
    ::= {attributeType 22}

TeletexTerminalIdentifier ::= SEQUENCE {
    teletexTerminalPrintableString
        (SIZE(1..ub-teletex-terminal-id)),
    parameters TeletexNonBasicParameters
        OPTIONAL}

facsimileTelephoneNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        FacsimileTelephoneNumber
    ::= {attributeType 23}

FacsimileTelephoneNumber ::= SEQUENCE {
    telephoneNumber PrintableString
        (SIZE(1..ub-telephone-number)),
    parameters G3FacsimileNonBasicParameters OPTIONAL}
```

```
x121Address ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                NumericString
                        (SIZE(1..ub-x121-address))
        MATCHES FOR EQUALITY SUBSTRINGS
        ::= {attributeType 24}

internationalISDNNumber ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                NumericString
                        (SIZE(1..ub-isdn-address))
        ::= {attributeType 25}

registeredAddress ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX PostalAddress
        ::= {attributeType 26}

destinationIndicator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                PrintableString
                        (SIZE(1..ub-destination-indicator))
        - alphabetical characters only
        MATCHES FOR EQUALITY SUBSTRINGS
        ::= {attributeType 27}

preferredDeliveryMethod ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                SEQUENCE OF INTEGER {
                        any-delivery-method     (0),
                        mhs-delivery            (1),
                        physical-delivery       (2),
                        telex-delivery          (3),
                        teletex-delivery        (4),
                        g3-facsimile-delivery   (5),
                        g4-facsimile-delivery   (6),
                        ia5-terminal-delivery   (7),
                        videotex-delivery       (8),
                        telephone-delivery      (9)}
        SINGLE VALUE
        ::= {attributeType 28}

presentationAddress ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                PresentationAddress
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= {attributeType 29}

PresentationAddress ::= SEQUENCE {
        pSelector       [0]   OCTET STRING OPTIONAL,
        sSelector       [1]   OCTET STRING OPTIONAL,
        tSelector       [2]   OCTET STRING OPTIONAL,
        nAddresses      [3]   SET SIZE (1..MAX) OF OCTET STRING}

supportedApplicationContext ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                objectIdentifierSyntax
        ::= {attributeType 30}

member ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                distinguishedNameSyntax
        ::= {attributeType 31}

owner ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                distinguishedNameSyntax
        ::= {attributeType 32}
```

```
roleOccupant ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                distinguishedNameSyntax
        ::= {attributeType 33}

seeAlso ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX
                distinguishedNameSyntax
        ::= {attributeType 34}

userPassword UserPassword
        ::= {attributeType 35}

userCertificate UserCertificate
        ::= {attributeType 36}

cACertificate CACertificate
        ::= {attributeType 37}

authorityRevocationList AuthorityRevocationList
        ::= {attributeType 38}

certificateRevocationList CertificateRevocationList
        ::= {attributeType 39}

CrossCertificatePair CrossCertificatePair
        ::= {attributeType 40}


-- sintaxis de atributo --

undefined ATTRIBUTE-SYNTAX
        ANY
        ::= {attributeSyntax 0}

distinguishedNameSyntax DistinguishedNameSyntax
        ::= {attributeSyntax 1}

objectIdentifierSyntax ObjectIdentifierSyntax
        ::= {attributeSyntax 2}

caseExactStringSyntax ATTRIBUTE-SYNTAX
        CHOICE {T61String, PrintableString}
        MATCHES FOR EQUALITY SUBSTRINGS
        ::= {attributeSyntax 3}

caseIgnoreSyntax ATTRIBUTE-SYNTAX
        CHOICE {T61String, PrintableString}
        MATCHES FOR EQUALITY SUBSTRINGS
        ::= {attributeSyntax 4}

printableStringSyntax ATTRIBUTE-SYNTAX
        PrintableString
        MATCHES FOR EQUALITY SUBSTRINGS
        ::= {attributeSyntax 5}

numericStringSyntax ATTRIBUTE-SYNTAX
        NumericString
        MATCHES FOR EQUALITY SUBSTRINGS
        ::= {attributeSyntax 6}

caseIgnoreListSyntax ATTRIBUTE-SYNTAX
        SEQUENCE OF
                CHOICE {T61String, PrintableString}
        MATCHES FOR EQUALITY SUBSTRINGS
        ::= {attributeSyntax 7}

booleanSyntax ATTRIBUTE-SYNTAX
        BOOLEAN
        MATCHES FOR EQUALITY
        ::= {attributeSyntax 8}
```

```
integerSyntax ATTRIBUTE-SYNTAX
        INTEGER
        MATCHES FOR EQUALITY ORDERING
        ::= {attributeSyntax 9}

octetStringSyntax ATTRIBUTE-SYNTAX
        OCTET STRING
        MATCHES FOR EQUALITY SUBSTRINGS ORDERING
        ::= {attributeSyntax 10}

uTCTimeSyntax ATTRIBUTE-SYNTAX
        UTCTime
        MATCHES FOR EQUALITY ORDERING
        ::= {attributeSyntax 11}

telephoneNumberSyntax ATTRIBUTE-SYNTAX
        PrintableString
                (SIZE(1..ub-telephone-number))
        MATCHES FOR EQUALITY SUBSTRINGS
        ::= {attributeSyntax 12}
```

## ANEXO B

(a la Recomendación X.520)

**Indice alfabético de tipos de atributo y sintaxis de atributo**

### TIPOS DE ATRIBUTO



### SINTAXIS DE ATRIBUTO



* Conocido y utilizado por la guía.

TIPOS DE ATRIBUTO

| | | |
|---|---|---|
| N | Nombre común | § 5.2.1 |
| | Nombre de Estado o provincia | § 5.3.2 |
| | Nombre de localidad | § 5.3.2 |
| | Nombre de objeto con alias * | § 5.1.2 |
| | Nombre de organización | § 5.4.1 |
| | Nombre de oficina de entrega física | § 5.6.4 |
| | Nombre de país | § 5.3.1 |
| | Nombre de unidad organizacional | § 5.4.2 |
| | Número de serie | § 5.2.3 |
| | Número de teléfono de facsímil | § 5.7.4 |
| | Número de teléfono | § 5.7.1 |
| | Número telex | § 5.7.2 |
| O | Ocupante de rol | § 5.10.3 |
| P | Par de certificados recíprocos | § 5.11.6 |
| | Propietario | § 5.10.2 |
| T | Título | § 5.4.3 |
| V | Véase también | § 5.10.4 |

* Conocido y utilizado por la guía.

## ANEXO C

(a la Recomendación X.520)

### Cotas superiores

Este anexo forma parte de la Recomendación.

```
UpperBounds    {joint-ISO-CCITT ds(5) modules(1)
               upperBounds(10)}
DEFINITIONS ::=
BEGIN

-- exporta todo --

ub-answerback                   INTEGER ::= 8
ub-common-name                  INTEGER ::= 64
ub-surname                      INTEGER ::= 64
ub-serial-number                INTEGER ::= 64
ub-locality-name                INTEGER ::= 128
ub-state-name                   INTEGER ::= 128
ub-street-address               INTEGER ::= 128
ub-organization-name            INTEGER ::= 64
ub-organizational-unit-name     INTEGER ::= 64
ub-title                        INTEGER ::= 64
```

```
    ub-description                  INTEGER ::= 1024
    ub-business-category            INTEGER ::= 128
    ub-postal-line                  INTEGER ::= 6
    ub-postal-string                INTEGER ::= 30
    ub-postal-code                  INTEGER ::= 40
    ub-post-office-box              INTEGER ::= 40
    ub-physical-office-name         INTEGER ::= 128
    ub-telex-number                 INTEGER ::= 14
    ub-country-code                 INTEGER ::= 4
    ub-teletex-terminal-id          INTEGER ::= 24
    ub-telephone-number             INTEGER ::= 32
    ub-x121-address                 INTEGER ::= 15
    ub-international-isdn-number    INTEGER ::= 16
    ub-destination-indicator        INTEGER ::= 128
    ub-user-password                INTEGER ::= 128
END
```

**Recomendación X.521**

## LA GUIA - CLASES DE OBJETO SELECCIONADAS [1)]

*(Melbourne, 1988)*

INDICE



---

1) La Recomendación X.521 y la norma ISO 9594-7, Information Processing Systems - Open Systems Interconnection - The Directory - Selected Object classes (Sistemas de procesamiento de información - Interconexión de sistemas abiertos - La Guía - Clases de objeto seleccionadas) fueron redactadas en estrecha colaboración y están técnicamente alineadas.

SECCION 1 - *Clases de objeto seleccionadas*

5 *Definición de conjuntos de atributos útiles*

5.1 Conjunto de atributos de telecomunicaciones
5.2 Conjunto de atributos postales
5.3 Conjunto de atributos de localización
5.4 Conjunto de atributos organizacionales

6 *Definición de clases de objeto seleccionadas*

6.1 Cumbre
6.2 Alias
6.3 País
6.4 Localidad
6.5 Organización
6.6 Unidad organizacional
6.7 Persona
6.8 Persona organizacional
6.9 Rol organizacional
6.10 Grupo de nombres
6.11 Persona residencial
6.12 Proceso de aplicación
6.13 Entidad de aplicación
6.14 ASG
6.15 Dispositivo

*Anexo A* - Clases de objeto seleccionadas en NSA.1

*Anexo B* - Formas de nombre y estructuras de AIG sugeridas

## 0 Introducción

0.1 Este documento, junto con los otros de la misma serie, ha sido producido para facilitar la interconexión de los sistemas de procesamiento de información para proporcionar servicios de guía (de abonados). El conjunto formado por todos estos sistemas y la información de guía en ellos contenida puede visualizarse como un todo integrado, denominado la *guía*. La información contenida en la guía, conocida colectivamente como la base de información de la guía (BIG), se utiliza típicamente para facilitar la comunicación entre, con, o sobre objetos tales como entidades de aplicación, personas, terminales y listas de distribución.

0.2 La guía desempeña un papel importante en la interconexión de sistemas abiertos, cuyo propósito es permitir, con un mínimo de acuerdos técnicos fuera de las normas de interconexión propiamente dichas, la interconexión de sistemas de procesamiento de información:

- de diferentes fabricantes;
- sometidos a diferentes gestiones;
- de diferentes grados de complejidad; y
- de diferentes fechas de construcción.

0.3 Esta Recomendación define (en la sección uno) un número de conjuntos de atributos y de clases de objeto que pueden ser útiles en un sector de aplicaciones de la guía.

0.4 El anexo A, que forma parte integrante de la Recomendación, proporciona un módulo NSA.1 que contiene todas las definiciones de valor y tipo que aparecen en este documento.

0.5 El anexo B, que no forma parte de la Recomendación, da algunas reglas comunes de estructura y denominación que pueden o no ser utilizadas por las autoridades administrativas.

## 1 Alcance y campo de aplicación

1.1 Esta Recomendación define cierto número de conjuntos de atributos y de clases de objeto seleccionados que pueden ser útiles en un sector de aplicaciones de la guía. La definición de un conjunto de atributos implica la identificación de los atributos que contiene, y facilita la definición de clases de objeto. La definición de una clase de objeto implica atribuirle opcionalmente un identificador de objeto, y listar un número de tipos de atributo que son significativos para los objetos de dicha clase. Estas definiciones son utilizadas por la autoridad administrativa responsable de la gestión de la información de la guía.

1.2 Cualquier autoridad administrativa puede definir sus propias clases o subclases de objeto para cualquier propósito.

*Nota 1* - Para estas definiciones puede o no utilizarse la notación especificada en la Recomendación X.501.

*Nota 2* - Se recomienda que se prefiera utilizar una clase de objeto definida en este documento, o una subclase derivada de la misma, a generar una nueva, siempre que la semántica resulte adecuada para la aplicación.

1.3 Las autoridades administrativas pueden soportar algunas o todas las clases de objeto seleccionadas y también pueden añadir clases de objeto.

Todas las autoridades administrativas soportarán las clases de objeto que utiliza la guía para sus propios fines (las clases de objeto cumbre, alias y ASG).

## 2 Referencias

Recomendación X.200 - Modelo de referencia de interconexión de sistemas abiertos para aplicaciones del CCITT (véase también ISO 7498)

Recomendación X.500 - La guía - Visión de conjunto de conceptos, modelos y servicios (véase también ISO 9594-1)

Recomendación X.501 - La guía - Modelos (véase también ISO 9594-2)

## 3 Definiciones y abreviaturas

3.1 *Definiciones relativas al modelo de referencia ISA*

Esta Recomendación utiliza el siguiente término definido en la Recomendación X.200:

a) *entidad de aplicación*;

b) *proceso de aplicación.*

3.2 *Definiciones relativas al modelo de la guía*

Esta Recomendación utiliza términos definidos en la Recomendación X.501.

a) *atributo*;
b) *tipo de atributo*;
c) *árbol de información de la guía (AIG)*;
d) *agente del sistema de guía (ASG)*;
e) *conjunto de atributos*;
f) *asiento*;
g) *nombre*;
h) *clase de objeto*;
i) *subclase.*

## 4 Notación

Las clases de objeto se definen en este documento utilizando una notación especial, definida como una macro NSA.1, **OBJECT-CLASS**, en la Recomendación X.501. Se utiliza un identificador de objeto (**objectClass**) 'genérico' para especificar los identificadores de objeto que se asignan a clases de objeto. Su definición puede encontrarse en el anexo B de dicha Recomendación.

Los conjuntos de atributos se definen en esta Recomendación utilizando una notación especial, definida como una macro NSA.1, **ATTRIBUTE-SET**, en la Recomendación X.501. Se utiliza un identificador de objeto (**attributeSet**) 'genérico' para especificar los identificadores de objeto que se asignan a definiciones de conjuntos de atributos. Su definición puede encontrarse en el anexo B de dicha Recomendación.

SECCION 1 - *Clases de objeto seleccionadas*

## 5 Definición de conjuntos de atributos útiles

### 5.1 *Conjunto de atributos de telecomunicaciones*

Este conjunto de atributos se utiliza para definir los que se utilizan comúnmente para las comunicaciones comerciales.

```
telecommunicationAttributeSet ATTRIBUTE-SET
    CONTAINS {
        facsimileTelephoneNumber,
        iSDNAddress,
        telephoneNumber,
        teletexTerminalIdentifier,
        telexNumber, X12Address,
        preferredDeliveryMethod,
        destinationIndicator,
        registeredAddress }
        ::= {attributeSet 0}
```

### 5.2 *Conjunto de atributos postales*

Este conjunto de atributos se utiliza para definir los que están directamente asociados a la entrega postal.

```
postalAttributeSet ATTRIBUTE-SET
    CONTAINS {
        physicalDeliveryOfficeName,
        postalAddress,
        postalCode,
        postOfficeBox,
        streetAddress}
        ::= {attributeSet 1}
```

### 5.3 *Conjunto de atributos de localización*

Este conjunto de atributos se utiliza para definir los que se utilizan comúnmente con propósitos de búsqueda para indicar la localización de un objeto.

```
localeAttributeSet ATTRIBUTE-SET
    CONTAINS {
        localityName,
        stateOrProvinceName,
        streetAddress}
        ::= {attributeSet 2}
```

### 5.4 *Conjunto de atributos organizacionales*

Este conjunto de atributos se utiliza para definir los atributos que una unidad organizacional u organización pueden poseer típicamente.

```
organizationalAttributeSet ATTRIBUTE-SET
    CONTAINS {
        description,
        localeAttributeSet,
        postalAttributeSet,
        telecommunicationAttributeSet,
        businessCategory,
        seeAlso,
        searchGuide,
        userPassword}
        ::= {attributeSet 3}
```

## 6 Definición de clases de objeto seleccionadas

### 6.1 *Cumbre*

La clase de objeto *cumbre*, de la cual todas las otras clases de objeto son subclases, está definida, excepto para la asignación de un identificador de objeto, en la Recomendación X.501.

**top Top ::= {objetClass 0}**

### 6.2 *Alias*

La clase de objeto *alias*, de la cual pueden derivarse clases de asientos de alias, está definida, excepto para la asignación de un identificador de objeto, en la Recomendación X.501.

**alias Alias ::= {objectClass 1}**

### 6.3 *País*

Una clase de objeto *país* se utiliza para definir los asientos de país en el AIG.

```
country OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        countryName}
    MAY CONTAIN {
        description,
        searchGuide}
    ::= {objectClass 2}
```

### 6.4 *Localidad*

La clase de objeto *localidad* se utiliza para definir localidad en el AIG.

```
locality OBJECT-CLASS
    SUBCLASS OF top
    MAY CONTAIN {
        description,
        localityName,
        stateOrProvinceName,
        searchGuide,
        seeAlso,
        streetAddress}
    ::= {objectClass 3}
```

Debe existir, cuando menos, un nombre de localidad o un nombre de estado o provincia.

### 6.5 *Organización*

La clase de objeto *organización* se utiliza para definir asientos de organización en el AIG.

```
organization OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        organizationName}
    MAY CONTAIN {
        organizationalAttributeSet}
    ::= {objectClass 4}
```

### 6.6 *Unidad organizacional*

La clase de objeto *unidad organizacional* se utiliza para definir asientos que representan subdivisiones de una organización.

```
organizationalUnit OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        organizationalUnitName}
    MAY CONTAIN {
        organizationalAttributeSet}
    ::= {objectClass 5}
```

6.7 *Persona*

La clase de objeto *persona* se utiliza para definir asientos que representan genéricamente personas.

```
person OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName,
        surname}
    MAY CONTAIN{
        description,
        seeAlso,
        telephoneNumber,
        userPassword}
    ::= {objectClass 6}
```

6.8 *Persona organizacional*

La clase de objeto *persona organizacional* se utiliza para definir asientos que representan una persona empleada por una organización o asociada con ella o en alguna otra forma importante.

```
organizationalPerson OBJECT-CLASS
    SUBCLASS OF person
    MAY CONTAIN {
        localeAttributeSet,
        organizationalUnitName,
        postalAttributeSet,
        telecommunicationAttributeSet,
        title}
    ::= {objectClass 7}
```

6.9 *Rol organizacional*

La clase de objeto *rol organizacional* se utiliza para definir asientos que representan un rol organizacional, es decir, una posición o rol (función) en una organización. Normalmente, se considera que un rol organizacional es desempeñado por una determinada persona organizacional. Sin embargo, a lo largo de su existencia, un rol organizacional puede ser satisfecho por cierto número de personas organizacionales diferentes y sucesivas. En general, un rol organizacional puede ser desempeñado por una persona o por una entidad (ente no humano).

```
organizationalRole OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localeAttributeSet,
        organizationalUnitName,
        postalAttributeSet,
        preferredDeliveryMethod,
        roleOccupant,
        seeAlso,
        telecommunicationAttributeSet}
    ::= {objectClass 8}
```

6.10 *Grupo de nombres*

La clase de objeto *grupo de nombres* se utiliza para definir asientos que representan un grupo no ordenado de nombres que, a su vez, representa objetos individuales u otros grupos de nombres. La calidad de miembro de un grupo es estática, es decir, se modifica explícitamente por acción administrativa, y no se determina dinámicamente cada vez que se menciona el grupo.

La calidad de miembro de un grupo puede reducirse a un conjunto de nombres de objetos individuales, reemplazando cada grupo por su membrecía. Este proceso puede ser realizado en forma recursiva hasta que todos los nombres de grupos constituyentes hayan sido eliminados y queden únicamente los nombres de los objetos individuales.

```
groupOfNames OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName,
        member}
    MAY CONTAIN {
        description,
        organizationName,
        organizationalUnitName,
        owner,
        seeAlso,
        businessCategory}
    ::= {objectClass 9}
```

6.11 *Persona residencial*

La clase de objeto *persona residencial* se utiliza para definir asientos que representan una persona en el entorno residencial.

```
residentialPerson OBJECT-CLASS
    SUBCLASS OF person
    MUST CONTAIN {
        localityName}
    MAY CONTAIN {
        localeAttributeSet,
        postalAttributeSet,
        preferredDeliveryMethod,
        telecommunicationAttributeSet,
        businessCategory}
    ::= {objectClass 10}
```

6.12 *Proceso de aplicación*

La clase de objeto *proceso de aplicación* se utiliza para definir asientos que representan procesos de aplicación. Un proceso de aplicación es un elemento dentro de un sistema abierto real que procesa información para una aplicación específica (véase la Recomendación X.200).

```
applicationProcess OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localityName,
        organizationalUnitName,
        seeAlso}
    ::= {objectClass 11}
```

6.13 *Entidad de aplicación*

La clase de objeto *entidad de aplicación* se utiliza para definir asientos que representan entidades de aplicación. Una entidad de aplicación consiste en los aspectos de un proceso de aplicación que son referentes a la ISA.

```
applicationEntity OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName,
        presentationAddress}
    MAY CONTAIN {
        description,
        localityName,
```

**organizationName,**
**organizationalUnitName,**
**seeAlso,**
**supportedApplicationContext}**
**::= {objectClass 12}**

*Nota* - Si la entidad de aplicación está representada como un objeto de la guía, que es distinto de un proceso de aplicación se utiliza el atributo CommonName para transportar el valor del calificador de entidad de aplicación.

6.14 *ASG*

La clase de objeto *ASG* se utiliza para definir asientos que representan ASG. El ASG se define en la Recomendación X.501.

**dSA OBJECT-CLASS**
**SUBCLASS OF applicationEntity**
**MAY CONTAIN {**
**knowledgeInformation}**
**::= {objectClass 13}**

6.15 *Dispositivo*

La clase de objeto *dispositivo* se utiliza para definir asientos que representan dispositivos. Un dispositivo es una unidad física capaz de comunicar, como un módem, un lector de disco, etc.

**device OBJECT-CLASS**
**SUBCLASS OF top**
**MUST CONTAIN {**
**commonName}**
**MAY CONTAIN {**
**description,**
**localityName,**
**organizationName,**
**organizationalUnitName,**
**owner,**
**seeAlso,**
**serialNumber}**
**::= {objectClass 14}**

*Nota* - Se debe incluir, por lo menos, uno de **localityName**, **serialNumber**, **owner**. La elección depende del tipo de dispositivo.

6.16 *Usuario de Autenticación Fuerte*

La clase de objeto *usuario de autenticación fuerte* se utiliza para definir asientos de objetos que participan en una autenticación fuerte, como se indica en la Recomendación X.509.

**strongAuthenticationUser OBJECT-CLASS**
**SUBCLASS OF top**
**MUST CONTAIN {userCertificate}**
**::= {objectClass 15}**

6.17 *Autoridad de certificación*

La clase de objeto *autoridad de certificación* se utiliza para definir asientos de objetos que actúan como autoridades de certificación, como se indica en la Recomendación X.509.

**certificationAuthority OBJECT-CLASS**
**SUBCLASS OF top**
**MUST CONTAIN {**
**cACertificate,**
**certificateRevocationList,**
**authorityRevocationList }**
**MAY CONTAIN {crossCertificatePair}**
**::= {objectClass 16}**

ANEXO A

(a la Recomendación X.521)

## Clases de objeto seleccionadas en NSA.1

Este anexo contiene todas las definiciones de valor y tipo NSA.1 utilizadas en esta Recomendación en forma del módulo NSA.1, **SelectObjectClasses.**

```
SelectedObjectClasses    {joint-ISO-CCITT ds(5) modules(1)
                          selectedObjectClasses(6)}
DEFINITIONS ::=
BEGIN

-- exporta todo

IMPORTS
    objectClass, attributeSet, informationFramework, selectedAttributeTypes
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}
    OBJECT-CLASS,ATTRIBUTE-SET, Top, Alias
        FROM InformationFramework informationFramework
    authorityRevocationList, businessCategory, CACertificate, certificateRevocationList,
    commonName, countryName, description, destinationIndicator, facsimileTelephoneNumber,
    internationalISDNNumber, knowledgeInformation, localityName, member, organizationName,
    organizationalUnitName, owner, physicalDeliveryOfficeName, postOfficeBox, postalAddress,
    postalCode, preferredDeliveryMethod, presentationAddress, registeredAddress,
    roleOccupant, searchGuide, seeAlso, serialNumber, stateOrProvinceName, streetAddress,
    supportedApplicationContext, surname, telephoneNumber, teletexTerminalIdentifier,
    telexNumber, title, userCertificate, userPassword, x121Address
        FROM  SelectedAttributeTypes selectedAttributeTypes;

telecommunicationAttributeSet ATTRIBUTE-SET
    CONTAINS {
        facsimileTelephoneNumber,
        iSDNAddress,
        telephoneNumber,
        teletexTerminalIdentifier,
        telexNumber,
        x121Address, preferredDeliveryMethod, destinationIndicator,
        registeredAddress}
        ::= {attributeSet 0}

postalAttributeSet ATTRIBUTE-SET
    CONTAINS {
        physicalDeliveryOfficeName,
        postalAddress,
        postalCode,
        postOfficeBox,
        streetAddress}
        ::= {attributeSet 1}

localeAttributeSet ATTRIBUTE-SET
    CONTAINS {
        localityName,
        stateOrProvinceName,
        streetAddress}
        ::= {attributeSet 2}

organizationalAttributeSet ATTRIBUTE-SET
    CONTAINS {
        description,
        localeAttributeSet,
        postalAttributeSet,
        telecommunicationAttributeSet,
        businessCategory,
```

```
            seeAlso,
            searchGuide,
            userPassword}
            ::= {attributeSet 3}

top         Top ::= {objectclass 0}

alias       Alias ::= {objectClass 1}

country OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            countryName}
      MAY CONTAIN {
            description,
            searchGuide}
      ::= {objectClass 2}

locality OBJECT-CLASS
      SUBCLASS OF top
      MAY CONTAIN {
            description,
            localityName,
            stateOrProvinceName,
            searchGuide,
            seeAlso,
            streetAddress}
      ::= {objectClass 3}

organization OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            organizationName}
      MAY CONTAIN {
            organizationalAttributeSet}
      ::= {objectClass 4}
organizationalUnit OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            organizationalUnitName}
      MAY CONTAIN {
            organizationalAttributeSet}
      ::= {objectClass 5}

person OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            commonName,
            surname}
      MAY CONTAIN {
            description,
            seeAlso,
            telephoneNumber,
            userPassword}
      ::= {objectClass 6}

organizationalPerson OBJECT-CLASS
      SUBCLASS OF person
      MAY CONTAIN {
            localeAttributeSet,
            organizationalUnitName,
            postalAttributeSet,
            telecommunicationAttributeSet,
            title}
      ::= {objectClass 7}
```

```
organizationalRole OBJECT-CLASS
        SUBCLASS OF top
        MUST CONTAIN {
                commonName}
        MAY CONTAIN {
                description,
                localeAttributeSet,
                organizationalUnitName,
                postalAttributeSet,
                preferredDeliveryMethod,
                roleOccupant,
                seeAlso,
                telecommunicationAttributeSet}
        ::= {objectClass 8}

groupOfNames OBJECT-CLASS
        SUBCLASS OF top
        MUST CONTAIN {
                commonName,
                member}
        MAY CONTAIN {
                description,
                organizationName,
                organizationalUnitName,
                owner,
                seeAlso,
                businessCategory}
        ::= {objectClass 9}

residentialPerson OBJECT-CLASS
        SUBCLASS OF person
        MUST CONTAIN {
                localityName}
        MAY CONTAIN {
                localeAttributeSet,
                postalAttributeSet,
                preferredDeliveryMethod,
                telecommunicationAttributeSet,
                businessCategory}
        ::= {objectClass 10}

applicationProcess OBJECT-CLASS
        SUBCLASS OF top
        MUST CONTAIN {
                commonName}
        MAY CONTAIN {
                description,
                localityName,
                organizationalUnitName,
                seeAlso}
        ::= {objectClass 11}

applicationEntity OBJECT-CLASS
        SUBCLASS OF top
        MUST CONTAIN {
                commonName,
                presentationAddress}
        MAY CONTAIN {
                description,
                localityName,
                organizationName,
                organizationalUnitName,
                seeAlso,
                supportedApplicationContext}
        ::= {objectClass 12}
```

```
dSA OBJECT-CLASS,
    SUBCLASS OF applicationEntity
    MAY CONTAIN {
        knowledgeInformation}
    ::= {objectClass 13}

device OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localityName,
        organizationName,
        organizationalUnitName,
        owner,
        seeAlso,
        serialNumber}
    ::= {objectClass 14}

strongAuthenticationUser OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        userCertificate}
    ::= {objectClass 15}

certificationAuthority OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        cACertificate,
        certificateRevocationList,
        authorityRevocationList}
    MAY CONTAIN {
        crossCertificatePair}
    ::=  {objectClass 16}

END
```

## ANEXO B

(a la Recomendación X.521)

### Formas de nombre y estructuras de AIG sugeridas

Este anexo no forma parte integrante de la Recomendación.

Este anexo sugiere algunas prácticas comunes de denominación y estructuras del AIG que pueden ser utilizadas o no por una autoridad administrativa. Las prácticas de denominación y las definiciones de estructura AIG para una clase de objeto incluyen especificaciones de los atributos utilizados para la denominación, y qué clases de objeto pueden tener su asiento superior o su asiento subordinado en el AIG. Todos los asientos de una clase de objeto dada tienen que incluir, cuando menos, los atributos utilizados para denominación. Los usuarios de la guía deben estar informados de las formas de nombre sugeridas, para poder imaginarse los nombres de los objetos con los que comunican. Los párrafos siguientes sugieren reglas de denominación y de estructura para algunas clases de objeto.

Las reglas de estructura se describen en la figura B-1/X.521.

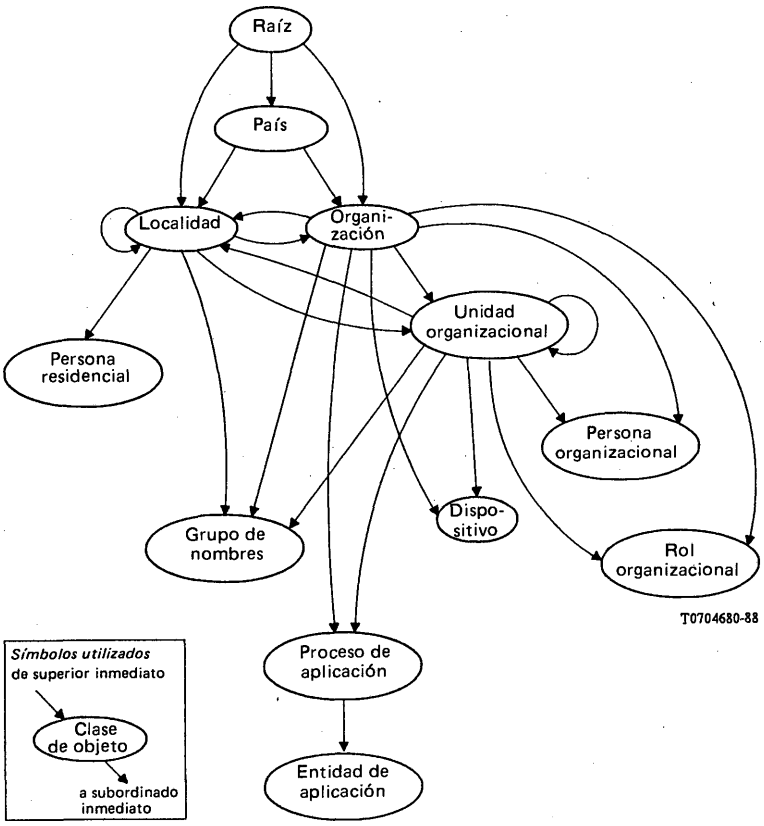



FIGURA B-1/X.521

**Estructura sugerida de AIG**

B.1 *País*

El atributo **countryName** se utiliza para denominación.

La raíz es el asiento inmediato superior de la clase de objeto **country**.

B.2 *Organización*

El atributo **organizationName** se utiliza para denominación.

Raíz, **country** o **locality** pueden ser asientos inmediatamente superiores a la clase de objeto **organization**.

*Nota* - El hecho de que la organización está directamente bajo la raíz significa que es una organización internacional. Los valores del atributo **organizationName** para organizaciones internacionales serán diferentes en cada caso.

B.3 *Localidad*

El atributo **localityName**, o **stateOrProvinceName**, se utiliza para denominación.

Raíz, **country**, **locality**, **organization** o **organizationalUnit** pueden ser asientos inmediatamente superiores a la clase de objeto **locality**.

B.4 *Unidad organizacional*

El atributo **organizationalUnitName** se utiliza para denominación.

**organization**, **organizationalUnit** o **locality** pueden ser asientos inmediatamente superiores a la clase de objeto **organizationalUnit**.

B.5 *Persona organizacional*

El atributo **commonName** y, opcionalmente, **organizationalUnitName** se utilizan para denominación.

**organization** o **organizationalUnit** pueden ser inmediatamente superiores a asientos de la clase de objeto **organizationalPerson.**

*Nota* - El atributo **organizationalUnitName** puede ser adquirido en nombres de dos modos: teniendo un objeto **organizationalUnit** como superior, o teniendo dicho atributo directamente.

B.6 *Rol organizacional*

El atributo **commonName** se utiliza para denominación.

**organization** o **organizationalUnit** pueden ser inmediatamente superiores a asientos de la clase de objeto **organizationalRole.**

*Nota* - El atributo **organizationalUnitName** puede ser adquirido en nombres de dos modos: teniendo un objeto **organizationalUnit** como superior, o teniendo dicho atributo directamente.

B.7 *Grupo de nombres*

El atributo **commonName** se utiliza para denominación.

**locality, organization** o **organizationalUnit** pueden ser inmediatamente superiores a asientos de la clase de objeto **groupOfNames.**

*Nota* - El atributo de **organizationalUnitName** puede ser adquirido en nombres de dos modos: teniendo el objeto **organizationalUnit** como superior, o teniendo dicho atributo directamente.

B.8 *Persona residencial*

Los atributos **commonName** y, opcionalmente **streetAddress** se utilizan para denominación.

**locality** es inmediatamente superior a asientos de clase de objeto **residentialPerson.**

B.9 *Entidad de aplicación*

El atributo **commonName** se utiliza para denominación. El **commonName** debe contener un calificador de entidad de aplicación (véase la Recomendación X.200).

El **applicationProcess** es inmediatamente superior a asientos de la clase de objeto **applicationEntity.**

B.10 *Dispositivo*

El atributo **commonName** se utiliza para denominación.

**organization** o **organizationalUnit** pueden ser inmediatamente superiores a asientos de la clase de objeto **device.**

*Nota* - El atributo **organizationalUnitName** puede ser adquirido en nombres de dos modos: teniendo un objeto **organizationalUnit** como superior, o teniendo dicho atributo directamente.

B.11 *Proceso de aplicación*

El atributo **commonName** se utiliza para denominación.

**organization** o **organizationalUnit** pueden ser inmediatamente superiores a asientos de la clase de objeto **applicationProcess.**

*Notas 1* - En la Recomendación X.200 se describe la manera de elegir un **commonName** para una entidad de aplicación.

*Nota 2* - El atributo de **organizationalUnitName** puede ser adquirido en nombre de dos modos: teniendo un objeto **organizationalUnit** como superior, o teniendo dicho atributo directamente.

ISBN 92-61-03733-X