



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**X.518**

(08/97)

SERIE X: REDES DE DATOS Y COMUNICACIÓN  
ENTRE SISTEMAS ABIERTOS

Directorio

---

**Tecnología de la información – Interconexión  
de sistemas abiertos – El directorio:  
Procedimientos para operación distribuida**

Recomendación UIT-T X.518

(Anteriormente Recomendación del CCITT)

---

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

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



**NORMA INTERNACIONAL 9594-4**

**RECOMENDACIÓN UIT-T X.518**

**TECNOLOGÍA DE LA INFORMACIÓN – INTERCONEXIÓN DE SISTEMAS  
ABIERTOS – EL DIRECTORIO: PROCEDIMIENTOS  
PARA OPERACIÓN DISTRIBUIDA**

**Resumen**

Esta Recomendación | Norma Internacional especifica los procedimientos en virtud de los cuales interfuncionan los componentes distribuidos de directorio para proporcionar un servicio coherente a sus usuarios.

**Orígenes**

La Recomendación UIT-T X.518 se aprobó el 9 de agosto de 1997. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 9594-4.

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT.

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

## NOTA

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

## PROPIEDAD INTELECTUAL

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

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

© UIT 2000

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

# ÍNDICE

	<i>Página</i>
SECCIÓN 1 – GENERALIDADES .....	1
1 Alcance .....	1
2 Referencias normativas.....	1
2.1 Recomendaciones   Normas Internacionales idénticas .....	1
3 Definiciones.....	2
3.1 Definiciones del modelo de referencia OSI.....	2
3.2 Definiciones básicas relativas al directorio .....	2
3.3 Definiciones relativas al modelo de directorio .....	2
3.4 Definiciones relativas al modelo de información del agente de sistema de directorio (DSA, <i>directory system agent</i> ) .....	3
3.5 Definiciones relativas a replicación de directorio .....	3
3.6 Definiciones relativas a la operación distribuida.....	3
4 Abreviaturas .....	5
5 Convenios .....	5
SECCIÓN 2 – VISIÓN DE CONJUNTO .....	6
6 Visión de conjunto.....	6
SECCIÓN 3 – MODELOS DE DIRECTORIO DISTRIBUIDO .....	7
7 Modelo de sistema de directorio distribuido.....	7
8 Modelo de interacciones de los DSA.....	8
8.1 Descomposición de una petición .....	8
8.2 Uniconcatenación .....	8
8.3 Multiconcatenación .....	9
8.4 Referimiento.....	10
8.5 Determinación del modo .....	11
SECCIÓN 4 – SERVICIO ABSTRACTO DE DSA .....	11
9 Visión de conjunto del servicio abstracto de DSA .....	11
10 Tipos de información.....	11
10.1 Introducción .....	11
10.2 Tipos de información definidos en otro lugar .....	12
10.3 Argumentos de concatenación.....	12
10.4 Resultados de concatenación .....	14
10.5 Avance (o progreso) de la operación .....	15
10.6 Información de rastreo.....	15
10.7 Tipo de referencia.....	16
10.8 Información de punto de acceso .....	16
10.9 Exclusiones.....	17
10.10 Referencia de continuación .....	18
11 Vincular y desvincular.....	19
11.1 DSA Bind (DSA vincular) .....	19
11.2 DSA Unbind (DSA desvincular).....	19
12 Operaciones concatenadas .....	19
12.1 Operaciones concatenadas.....	19
12.2 Operación ChainedAbandon .....	20
12.3 Operaciones concatenadas y versión de protocolo .....	21

13	Errores concatenados.....	21
13.1	Introducción.....	21
13.2	Remisión de DSA.....	21
SECCIÓN 5 – PROCEDIMIENTOS DISTRIBUIDOS.....		22
14	Introducción.....	22
14.1	Alcance y límites.....	22
14.2	Conformidad.....	22
14.3	Modelo conceptual.....	22
14.4	Operación individual y cooperativa de DSA.....	22
14.5	Acuerdos de cooperación entre los DSA.....	23
15	Comportamiento de directorio distribuido.....	23
15.1	Realización cooperativa de operaciones.....	23
15.2	Fases del procesamiento de operaciones.....	23
15.3	Gestión de operaciones distribuidas.....	24
15.4	Tratamiento de bucles.....	25
15.5	Otras consideraciones relativas a la operación distribuida.....	26
15.6	Autenticación de operaciones distribuidas.....	27
16	El despachador de operaciones.....	28
16.1	Conceptos generales.....	29
16.2	Procedimientos del despachador de operaciones.....	31
16.3	Visión de conjunto de los procedimientos.....	32
17	Validación de petición.....	34
17.1	Introducción.....	34
17.2	Parámetros del procedimiento.....	35
17.3	Definición del procedimiento.....	35
18	Resolución de nombre.....	37
18.1	Introducción.....	37
18.2	Parámetros del procedimiento Find DSE.....	37
18.3	Procedimientos.....	38
19	Evaluación de la operación.....	46
19.1	Procedimiento de modificación.....	47
19.2	Procedimiento de interrogación de una sola inserción.....	53
19.3	Procedimiento de interrogación de múltiples inserciones.....	54
20	Procedimientos de referencia de continuación.....	63
20.1	Estrategia de concatenación en presencia de sombreado.....	63
20.2	Envío de subpeticiones concatenadas a un DSA distante.....	66
20.3	Parámetros del procedimiento.....	67
20.4	Definición de los procedimientos.....	67
20.5	Procedimiento Abandon (abandonar).....	74
21	Procedimiento Results Merging (fusión de resultados).....	76
22	Procedimientos para autenticación distribuida.....	77
22.1	Autenticación de originador.....	77
22.2	Autenticación de resultados.....	79
SECCIÓN 6 – ADMINISTRACIÓN DE CONOCIMIENTO.....		79
23	Visión de conjunto de la administración de conocimiento.....	79
23.1	Mantenimiento de referencias de conocimiento.....	79
23.2	Petición de referencia de remisión.....	81
23.3	Inconsistencias de conocimiento.....	81
23.4	Referencias de conocimiento y contextos.....	82

	<i>Página</i>
24 Vinculaciones operacionales jerárquicas .....	82
24.1 Características de tipo vinculación operacional .....	83
24.2 Definición de la clase de objeto de información vinculación operacional .....	85
24.3 Procedimientos DSA para gestión de vinculaciones operacionales .....	86
24.4 Procedimientos para operaciones .....	90
24.5 Utilización de contextos de aplicación .....	90
25 Vinculación operacional jerárquica no específica .....	90
25.1 Características del tipo vinculación operacional .....	90
25.2 Definición de la clase de objeto de información vinculación operacional .....	91
25.3 Procedimientos de DSA para gestión de vinculaciones operacionales jerárquicas no específicas.....	92
25.4 Procedimientos para operaciones .....	94
25.5 Utilización de contextos de aplicación .....	94
Anexo A – ASN.1 para operaciones distribuidas .....	95
Anexo B – Ejemplo de resolución de nombre distribuida .....	98
Anexo C – Uso distribuido de la autenticación .....	100
C.1 Resumen .....	100
C.2 Modelo de protección distribuida .....	100
C.3 Operaciones concatenadas firmadas .....	101
C.4 Operaciones concatenadas criptadas .....	102
C.5 Operaciones distribuidas firmadas y criptadas .....	105
Anexo D – Especificación de tipos de vinculaciones operacionales jerárquicas y jerárquicas no específicas .....	107
Anexo E – Ejemplo de mantenimiento de conocimiento .....	109
Anexo F – Enmiendas y correcciones .....	112



## Introducción

Esta Recomendación | parte de Norma Internacional, junto con otras Recomendaciones | Normas Internacionales, ha sido elaborada para facilitar la interconexión de sistemas de procesamiento de información con el fin de proporcionar servicios de directorio. El conjunto de todos estos sistemas, junto con la información de directorio que contienen, puede considerarse como un todo integrado, llamado el *directorio*. La información contenida por el directorio, denominada colectivamente base de información de directorio (DIB, *directory information base*), se utiliza típicamente para facilitar la comunicación entre, con o sobre objetos tales como entidades de aplicación, personas, terminales y listas de distribución.

El directorio desempeña un papel importante en la interconexión de sistemas abiertos (OSI), cuyo objetivo 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.

En esta Recomendación | Norma Internacional se especifican los procedimientos según los cuales los componentes distribuidos de directorio interfuncionan para proporcionar un servicio coherente a sus usuarios.

Esta tercera edición revisa y mejora técnicamente la segunda edición de la presente Recomendación | Norma Internacional, pero no la substituye. Las implementaciones pueden seguir alegando conformidad con la segunda edición. Sin embargo, en algún punto, no se admitirá la segunda edición (es decir, los defectos informados ya no serán resueltos). Se recomienda que las implementaciones se conformen con esta tercera edición lo antes posible.

Esta tercera edición especifica las versiones 1 y 2 de los protocolos de directorio.

Las ediciones primera y segunda especifican también la versión 1. La mayor parte de los servicios y protocolos especificados en esta edición están diseñados para funcionar según la versión 1. Cuando se ha negociado la versión 1, las diferencias entre los servicios y entre los protocolos definidos en las tres ediciones se acomodan utilizando las reglas de extensibilidad definidas en esta edición de la Rec. UIT-T X.519 | ISO/CEI 9594-5. No obstante, algunos servicios y protocolos mejorados, por ejemplo, los errores signados, no funcionarán a menos que todas las entidades de directorio que participan en la operación hayan negociado la versión 2.

Los implementadores deben observar que existe un proceso de resolución de defectos y que se pueden aplicar correcciones a esta Norma Internacional en forma de corrigenda técnicos. Se aplicarán correcciones idénticas a esta Recomendación en forma de corrigenda y/o una guía del implementador. Se puede obtener de la Secretaría del subcomité una lista de los corrigenda técnicos aprobados para esta Norma Internacional. Los corrigenda técnicos publicados están disponibles en las organizaciones nacionales de normalización. Los corrigenda UIT-T y la guía del implementador pueden obtenerse en el sitio Web de la UIT.

El anexo A, que es parte integrante de esta Recomendación | Norma Internacional, proporciona el módulo ASN.1 para operaciones distribuidas de directorio.

El anexo B, que no es parte integrante de esta Recomendación | Norma Internacional, presenta un ejemplo de resolución de nombre distribuida.

El anexo C, que no es parte integrante de esta Recomendación | Norma Internacional, describe la autenticación en el entorno de operaciones distribuidas.

El anexo D, que es parte integrante de esta Recomendación | Norma Internacional, proporciona las definiciones de las clases de objeto de información ASN.1 introducidas en esta Especificación de directorio.

El anexo E, que no es parte integrante de esta Recomendación | Norma Internacional, explica el mantenimiento de conocimiento.

El anexo F, que no es parte integrante de esta Recomendación | Norma Internacional, contiene una reseña de las enmiendas e informes de defectos que se han incorporado en la presente edición de esta Recomendación | Norma Internacional.

## NORMA INTERNACIONAL

## RECOMENDACIÓN UIT-T

**TECNOLOGÍA DE LA INFORMACIÓN – INTERCONEXIÓN DE SISTEMAS  
ABIERTOS – EL DIRECTORIO: PROCEDIMIENTOS  
PARA OPERACIÓN DISTRIBUIDA**

## SECCIÓN 1 – GENERALIDADES

**1 Alcance**

La presente Recomendación | Norma Internacional especifica el comportamiento de los DSA que participan en la aplicación directorio distribuido. El comportamiento permitido está concebido de manera tal que se asegure un servicio coherente dada una amplia distribución de la DIB a través de muchos DSA.

No se pretende que el directorio 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 normativas**

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

**2.1 Recomendaciones | Normas Internacionales idénticas**

- Recomendación UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de referencia básico: El modelo básico.*
- Recomendación UIT-T X.500 (1997) | ISO/CEI 9594-1:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Visión de conjunto de conceptos, modelos y servicios.*
- Recomendación UIT-T X.501 (1997) | ISO/CEI 9594-2:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Modelos.*
- Recomendación UIT-T X.509 (1997) | ISO/CEI 9594-8:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Marco de autenticación.*
- Recomendación UIT-T X.511 (1997) | ISO/CEI 9594-3:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Definición de servicio abstracto.*
- Recomendación UIT-T X.519 (1997) | ISO/CEI 9594-5:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Especificaciones de protocolo.*

- Recomendación UIT-T X.520 (1997) | ISO/CEI 9594-6:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Tipos de atributos seleccionados.*
- Recomendación UIT-T X.521 (1997) | ISO/CEI 9594-7:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Clases de objetos seleccionadas.*
- Recomendación UIT-T X.525 (1997) | ISO/CEI 9594-9:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Replicación.*
- Recomendación UIT-T X.530 (1997) | ISO/CEI 9594-10:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Utilización de la gestión de sistemas para la administración del directorio.*
- Recomendación UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- Recomendación UIT-T X.681 (1997) | ISO/CEI 8824-2:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*
- Recomendación UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.*
- Recomendación UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de las especificaciones de la notación de sintaxis abstracta uno.*
- Recomendación UIT-T X.880 (1994) | ISO/CEI 13712-1:1995, *Tecnología de la información – Operaciones a distancia: Conceptos, modelo y notación más corrigendum técnico 1 (1995).*
- Recomendación UIT-T X.880 (1994)/Enm.1 (1995) | ISO/CEI 13712 -1:1995/Enm.1:1996, *Tecnología de la información – Operaciones a distancia: Conceptos, modelo y notación – Enmienda 1: Operaciones incorporadas.*
- Recomendación UIT-T X.881 (1994) | ISO/CEI 13712-2:1995, *Tecnología de la información – Operaciones a distancia – Realizaciones de interconexión de sistemas abiertos: Definición de servicio del elemento de servicio de operaciones a distancia.*
- Recomendación UIT-T X.881 (1994)/Enm.1 (1995) | ISO/CEI 13712-2:1995/Enm.1:1996, *Tecnología de la información – Operaciones a distancia: Realizaciones de interconexión de sistemas abiertos – Definición del servicio del elemento de servicio de operaciones a distancia – Enmienda 1: Correspondencia de A-DATOS-UNIDAD y operaciones incorporadas.*

### 3 Definiciones

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

#### 3.1 Definiciones del modelo de referencia OSI

El siguiente término está definido en la Rec. UIT-T X.200 | ISO/CEI 7498-1:

- *título de entidad de aplicación.*

#### 3.2 Definiciones básicas relativas al directorio

Los siguientes términos están definidos en la Rec. UIT-T X.500 | ISO/CEI 9594-1:

- a) *(el) directorio;*
- b) *base de información de directorio.*

#### 3.3 Definiciones relativas al modelo de directorio

Los siguientes términos están definidos en la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- a) *punto de acceso;*
- b) *alias;*
- c) *nombre distinguido;*

- d) *árbol de información de directorio*;
- e) *agente de sistema de directorio*;
- f) *agente de usuario de directorio*;
- g) *nombre distinguido relativo*.

### 3.4 Definiciones relativas al modelo de información del agente de sistema de directorio (DSA, *directory system agent*)

Los siguientes términos están definidos en la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- a) *categoría*;
- b) *utilizable en común*;
- c) *prefijo de contexto*;
- d) *referencia de remisión (remisión, referencia cruzada)*;
- e) *fragmento de DIB*;
- f) *árbol de información de DSA*;
- g) *inserción (o asiento) específica del DSA (DSE, directory specific entry)*;
- h) *tipo de DSE*;
- i) *referencia superior inmediata*;
- j) *información de conocimiento*;
- k) *categoría de referencia de conocimiento*;
- l) *tipo de referencia de conocimiento*;
- m) *contexto de denominación*;
- n) *conocimiento no específico*;
- o) *referencia subordinada no específica*;
- p) *atributo operacional*;
- q) *trayecto (o camino) de referencia*;
- r) *conocimiento específico*;
- s) *referencia subordinada*;
- t) *referencia superior*.

### 3.5 Definiciones relativas a replicación de directorio

Los siguientes términos están definidos en la Rec. UIT-T X.525 | ISO/CEI 9594-9:

- a) *compleción de atributo*;
- b) *vinculación operacional de sombreado (o de sombreo)*;
- c) *compleción subordinada*;
- d) *unidad de replicación*.

### 3.6 Definiciones relativas a la operación distribuida

A los efectos de esta Recomendación | Norma Internacional, se definen las siguientes definiciones.

**3.6.1 objeto de base; objeto básico:** La inserción de objeto o alias que es el objetivo de una operación, iniciada por el originador.

**3.6.2 concatenación; encadenamiento:** Término genérico para uniconcatenación o multiconcatenación.

**3.6.3 información de prefijo de contexto:** Información operacional y de usuario suministrada por el DSA superior al DSA subordinado en una RHOB con relación a vértices del DIT superiores al prefijo de contexto subordinado.

- 3.6.4 resolución de nombres distribuida:** El proceso en virtud del cual se efectúa la resolución de nombres en más de un DSA.
- 3.6.5 error:** Información enviada desde el ejecutante al peticionario y que transporta un resultado final negativo de una petición anteriormente recibida.
- 3.6.6 error duro:** Un error definitivo que indica que, en ese momento, la operación no puede ejecutarse sin una intervención externa.
- 3.6.7 vinculación operacional jerárquica (HOB, *hierarchical operational binding*):** Relación entre dos DSA maestros que contienen los mismos contextos de denominación, uno de los cuales está inmediatamente subordinado al otro, y en la cual el DSA superior contiene una referencia subordinada al DSA subordinado.
- 3.6.8 operaciones de modificación:** Estas son las operaciones modificación (modificar) de directorio, a saber, Modify Entry (modificar inserción), Add Entry (añadir inserción), Remove Entry (suprimir inserción), y ModifyDN (modificar DN).
- 3.6.9 multiconcatenación; multiencadenamiento:** Un modo de interacción en el cual un DSA que está procesando una petición envía, múltiples peticiones, ya sea en paralelo o secuencialmente, a un conjunto de otros DSA.
- 3.6.10 operaciones de interrogación de múltiples inserciones:** Estas son las operaciones de búsqueda en el directorio, a saber, List (listar) y Search (buscar).
- 3.6.11 resolución de nombres:** El proceso de localizar una inserción concordando secuencialmente con cada RDN en un nombre contemplado, a un vértice del DIT.
- 3.6.12 vinculación operacional jerárquica no específica (NHOB, *non-specific hierarchical operational binding*):** Relación entre dos DSA maestros que contienen contextos de denominación, uno de los cuales está inmediatamente subordinado al otro, y en la cual el DSA superior contiene una referencia subordinada no específica al DSA subordinado.
- 3.6.13 descomposición de referencia subordinada no específica (NSSR, *non-specific subordinate reference*):** Descomposición de referencias de conocimiento no específicas en subpeticiones que son continuadas por otros DSA; con respecto a estas subpeticiones se puede proceder de varias formas: el DSA que efectúa la descomposición puede concatenarlas a estos DSA, o se puede retornar una referencia de continuación que identifique los DSA al solicitante para que éste prosiga, o el DSA que efectúa la descomposición puede proseguir algunas de las subpeticiones y dejar otras inexploradas para que el solicitante las prosiga.
- 3.6.14 progreso (o avance) de la operación:** Un conjunto de valores que indican hasta qué punto ha tenido lugar la resolución de nombre.
- 3.6.15 originador:** El DUA que ha iniciado una operación (distribuida) específica.
- 3.6.16 efectuador:** DSA que recibe una petición (de efectuar una operación).
- 3.6.17 procedimiento:** Una especificación (informal) de la manera en que un DSA se corresponde con un conjunto dado de argumentos de entrada y su árbol de información de DSA a un resultado.
- NOTA – Los argumentos de entrada y los resultados deben corresponder a información recibida en una operación pedida e información enviada en una respuesta, o pueden representar etapas intermedias en la computación de una respuesta a partir de una operación pedida. En 14.2, la primera de estas modalidades de argumentos de entrada y resultados se denomina externos.
- 3.6.18 vinculación operacional jerárquica relevante (RHOB, *relevant hierarchical operational binding*):** O bien una HOB o una NHOB, según el contexto.
- 3.6.19 referimiento:** Un resultado final que puede ser retornado por un DSA que no puede efectuar una operación por sí mismo, y que identifica a otro y otros DSA más capaces para efectuar esa operación.
- 3.6.20 respuesta:** Un resultado o un error.
- 3.6.21 petición:** Información que consiste en un código de operación y argumentos asociados, para transportar una operación de directorio desde un peticionario a un efectuador.
- 3.6.22 descomposición de peticiones:** Descomposición de una petición en subpeticiones para que sean ejecutadas por otros DSA; estas subpeticiones pueden ser concatenadas a estos DSA por el DSA que efectúa la descomposición, o bien se puede retornar referencias de continuación que identifican los DSA, al peticionario, para que éste prosiga, o el DSA que efectúa la descomposición puede proseguir algunas de las subpeticiones y dejar otras inexploradas para que sea el peticionario quien las prosiga.

- 3.6.23 petionario:** Un DUA o DSA que envía una petición de efectuar una operación (es decir, invocar una operación).
- 3.6.24 operaciones de interrogación de una sola inserción:** Operaciones de lectura del directorio, es decir, Read (leer) y Compare (comparar).
- 3.6.25 error blando:** Un error que puede ser transitorio o que puede indicar un problema localizado, en cuyo caso la utilización de una referencia de conocimiento o de un punto de acceso diferente puede permitir la obtención de un resultado o de un error duro.
- 3.6.26 agente de sistema de directorio subordinado:** De los dos DSA que comparten una HOB o una NHOB, el DSA que contiene el contexto de denominación subordinado.
- 3.6.27 subpetición:** Una petición generada por la descomposición de una petición.
- 3.6.28 agente de sistema de directorio superior:** El DSA que, entre los dos que comparten una vinculación operacional jerárquica, o jerárquica no específica, contiene el contexto de denominación superior.
- 3.6.29 agente de sistema de directorio superior, subordinado:** Dos DSA maestros que contienen los mismos contextos de denominación, uno de los cuales está inmediatamente subordinado al otro; la relación entre los dos DSA es manejada explícitamente vía una HOB (o una NHOB), o existe implícitamente en virtud del DSA superior que contiene una referencia subordinada (o subordinada no específica) al DSA subordinado.
- 3.6.30 nombre de objetivo de objeto:** El nombre de una inserción sobre la cual debe efectuarse la operación en una etapa dada de la resolución de nombre, o que participa en la evaluación de la operación.
- 3.6.31 uniconcatenación; monoconcatenación:** Un modo de interacción utilizado opcionalmente por un DSA que no puede efectuar una operación por sí mismo. El DSA *concatena* invocando una operación de otro DSA y enviando el resultado final al petionario original.

## 4 Abreviaturas

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

ASN.1	Notación de sintaxis abstracta uno ( <i>abstract syntax notation one</i> )
DISP	Protocolo de sombreado de información de directorio ( <i>directory information shadowing protocol</i> )
DMD	Dominio de gestión de directorio ( <i>directory management domain</i> )
DOP	Protocolo de gestión de vinculaciones operacionales de directorio ( <i>directory operational binding management protocol</i> )
DSE	Inserción (o asiento) específica de DSA ( <i>DSA specific entry</i> )
HOB	Vinculación operacional jerárquica ( <i>hierarchical operational binding</i> )
NHOB	Vinculación operacional jerárquica no específica ( <i>non-specific hierarchical operational binding</i> )
NSSR	Referencia subordinada no específica ( <i>non-specific subordinate reference</i> )
RHOB	Vinculación operacional jerárquica relevante ( <i>relevant hierarchical operational binding</i> )

## 5 Convenios

Con pequeñas excepciones, esta Especificación de directorio se ha preparado con arreglo a las directrices de "Presentación de textos comunes UIT-T | ISO/CEI" que figuran en la Guía para la cooperación entre el UIT-T y el JTC 1 de la ISO/CEI.

El término "Especificación de directorio" (como en "esta Especificación de directorio") se entenderá en el sentido de la Rec. UIT-T X.518 | ISO CEI 9594-4. El término "Especificaciones de directorio" se entenderá que designa a todas las Recomendaciones de la serie X.500 y todas las partes de ISO/CEI 9594.

Esta Especificación de directorio utiliza el término "sistemas de la edición de 1988" para hacer referencia a los sistemas conformes a la primera edición (1988) de las Especificaciones de directorio, es decir, la edición de 1988 de las Recomendaciones CCITT de la serie X.500 y la edición de ISO/CEI 9594:1990. Esta Especificación de directorio utiliza el término "sistemas de la edición de 1993" para hacer referencia a los sistemas conformes a la segunda edición (1993) de las Especificaciones de directorio, es decir, la edición de 1993 de las Recomendaciones UIT-T de la serie X.500 y la edición de ISO/CEI 9594:1995. Los sistemas conformes a esta tercera edición de las Especificaciones de directorio se indican como "sistemas de la edición 1997".

Esta Especificación de directorio presenta la notación ASN.1 en caracteres Helvética en negritas. Cuando se hace referencia a los tipos y valores ASN.1 en el texto normal, se los diferencia del texto normal presentándolos en caracteres Helvética en negritas. Para diferenciar los nombres de los procedimientos, a los que en general se hace referencia cuando se especifica la semántica del procesamiento, se los presenta en caracteres Times en negritas. Los permisos de control de acceso se presentan en caracteres Times en cursivas.

Si los elementos de una lista están numerados (en contraste con la utilización de "-" o de letras), se considerarán etapas de un procedimiento.

Esta Especificación de directorio define operaciones de directorio utilizando la notación de operación a distancia definida en la Rec. UIT-T X.880 | ISO/CEI 13712-1.

## SECCIÓN 2 – VISIÓN DE CONJUNTO

### 6 Visión de conjunto

El servicio abstracto de directorio permite la interrogación, recuperación y modificación de la información de directorio en la DIB. Este servicio se describe en términos del objeto abstracto de directorio según se especifica en la Rec. UIT-T X.511 | ISO/CEI 9594-3.

Por necesidad, la especificación del objeto de directorio abstracto no se refiere en modo alguno a la realización física del directorio: no se ocupa, en concreto, de la especificación de los agentes de sistema de directorio (DSA, *directory system agents*), en los que se almacena y gestiona la DIB, y a través de los que se presta el servicio. Más aún, esa especificación tampoco tiene en cuenta si la DIB es centralizada, es decir, contenida en un único DSA o distribuida entre un cierto número de DSA. Por consiguiente, las exigencias para que los DSA tengan conocimiento de, naveguen hacia, y cooperen con otros DSA a fin de facilitar el servicio abstracto en un entorno distribuido quedan también fuera del alcance de la descripción del servicio.

En esta Especificación de directorio se especifica el refinamiento del objeto de directorio abstracto, viniendo expresado ese refinamiento en base a uno o más objetos de DSA que en conjunto constituyen el servicio de directorio distribuido.

Se especifican además las maneras permitidas de distribuir la DIB entre uno o más DSA. En el caso límite en que la DIB estuviera contenida en un solo DSA, el directorio sería de hecho un directorio centralizado; para el caso en que la DIB esté distribuida entre dos o más DSA, se especifican mecanismos de conocimiento y navegación que garantizan que la totalidad de la DIB es potencialmente accesible desde todos los DSA que contengan inserciones constituyentes.

Partes de la DIB también pueden ser replicadas en múltiples objetos de DSA. Los protocolos descritos en esta Especificación de directorio contemplan el uso de la replicación para mejorar la accesibilidad, rendimiento y eficiencia del servicio de directorio distribuido. El uso de la información replicada está en cierta medida bajo control del usuario, mediante el uso de opciones de control del servicio. Los procedimientos descritos en esta Especificación de directorio también señalan algunas de las optimizaciones de diseño con el uso de la información replicada.

Asimismo, se especifican interacciones para el tratamiento de solicitudes que permiten el que determinadas características operativas de directorio sean controladas por sus usuarios. En concreto, el usuario controla si un DSA, en respuesta a una indagación de directorio relativa a información contenida en otro u otros DSA, puede interrogar al otro u otros DSA directamente (concatenación), o si debe responder con información sobre otro u otros DSA que puedan hacer avanzar la investigación (referimiento).

Por lo general, la elección de un DSA entre concatenar o referir vendrá determinada por los controles de servicio establecidos por el usuario, y por las circunstancias administrativas, operacionales o técnicas propias del DSA.

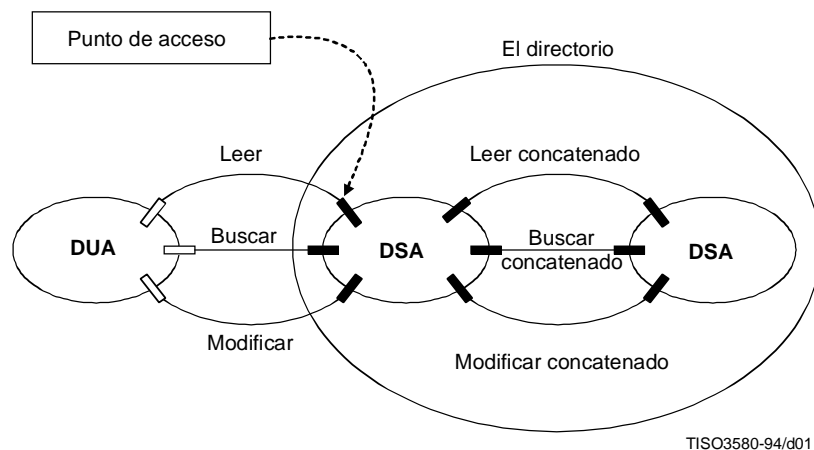
Habida cuenta de que, por lo general, el directorio será distribuido y que las consultas de directorio serán satisfechas por un número arbitrario de DSA cooperantes, que podrán arbitrariamente concatenar o referir, según los criterios expuestos más arriba, en esta Especificación de directorio se especifican los procedimientos apropiados a que deben atenerse los DSA en respuesta a consultas de un directorio distribuido. Dichos procedimientos garantizarán a los usuarios del servicio de directorio distribuido un servicio a la vez cómodo y coherente.

## SECCIÓN 3 – MODELOS DE DIRECTORIO DISTRIBUIDO

### 7 Modelo de sistema de directorio distribuido

El servicio abstracto de directorio, tal como se define en la Rec. UIT-T X.511 | ISO/CEI 9594-3, modela el directorio como un objeto que presta un conjunto de servicios de directorio a sus usuarios. Los usuarios del directorio ganan acceso a sus servicios a través de un punto de acceso. El directorio 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.

La figura 1 ilustra el modelo de directorio distribuido que servirá de base para especificar los aspectos distribuidos de directorio. Ilustra la composición de directorio (como un conjunto de uno o más DSA).



**Figura 1 – Objetos del modelo de directorio distribuido**

Los DSA se especifican con detalle en las cláusulas posteriores de esta Especificación de directorio. Esta cláusula meramente indica algunas de sus características a modo de introducción y para establecer la relación entre la Especificación de directorio y las demás Especificaciones de directorio.

Los DSA se definen para que la distribución de la DIB pueda acomodarse y para que diversos DSA situados a una gran distancia puedan interactuar de una manera cooperativa y regulada para ofrecer servicios de directorio a los usuarios de directorio (DUA).

La figura 1 ilustra la relación entre el servicio abstracto de directorio y el servicio abstracto de DSA. El servicio abstracto de directorio definido en la Rec. UIT-T X.511 | ISO/CEI 9594-3 se proporciona a través de cierto número de operaciones de directorio. Para realizar este servicio, los DSA que constituyen el directorio interactúan uno con otro. La naturaleza de esta interacción se define en términos del servicio que un DSA puede proporcionar al otro, el servicio abstracto de DSA. El servicio abstracto de DSA se proporciona a través de cierto número de operaciones, denominadas operaciones concatenadas, cada una de las cuales tiene su contrapartida en el servicio abstracto de directorio. Así, una operación dada en el servicio abstracto de directorio, por ejemplo leer (read) puede requerir que el DSA que proporciona el servicio abstracto interactúe con uno o más DSA que utilizan operaciones concatenadas, por ejemplo leer concatenado (Chained Read).



## 8 Modelo de interacciones de los DSA

Una característica básica de directorio es la de que, dada una DIB distribuida, un usuario deba poder obtener la satisfacción de cualquier petición de servicio (de acuerdo con las políticas de seguridad, de control de acceso y de administración) cualquiera que sea el punto de acceso en el que se origine la petición. El cumplimiento de este requisito exige que todo DSA que participe en la tramitación de una determinada solicitud de servicio tenga cierto conocimiento (tal como se especifica en la Rec. UIT-T X.501 | ISO/CEI 9594-2) de dónde se halla la información pedida y retorne ese conocimiento al peticionario o trate en su nombre de que se tramite la petición. (El peticionario puede ser un DUA u otro DSA; en el segundo caso, ambos DSA deben soportar el DSP.)

Para satisfacer estas exigencias se definen tres modos de interacción del DSA, a saber: "uniconcatenación" ("uni-chaining"), "multiconcatenación" ("multi-chaining"), y "referimiento" ("referral"). En todo el resto de esta Especificación de directorio se utilizará el término genérico concatenación para hacer referencia a la uniconcatenación y/o multiconcatenación, según proceda en el contexto. "Concatenación" hace referencia a la tentativa de un DSA de satisfacer una petición enviando una o más operaciones concatenadas a otros DSA; "referimiento" implica el retorno de información de conocimiento al peticionario, que podrá interactuar por sí mismo con el DSA o los DSA identificados en la información de conocimiento.

Una misma petición puede dar lugar a una interacción de uniconcatenación o de referimiento. Como otra posibilidad, la petición puede descomponerse en varias subpeticiones antes de la interacción. Como resultado de la descomposición de una petición pueden producirse interacciones de multiconcatenación o de referimiento, o una combinación de ambas. Se definen dos tipos de descomposición: descomposición de NSSR y descomposición de petición.

### 8.1 Descomposición de una petición

#### 8.1.1 Descomposición de NSSR

La descomposición de NSSR es el proceso de preparar peticiones idénticas, listas para ser transferidas (secuencialmente o en paralelo) a varios DSA subordinados como consecuencia de haberse encontrado una NSSR durante una resolución de nombre. Las referencias subordinadas no específicas no contienen los RDN de los contextos de denominación subordinados referenciados, por lo cual el DSA referenciante no puede saber qué DSA subordinado contiene qué contexto(s) de denominación subordinados. Durante la resolución de nombres, un DSA que encuentra NSSR enviará una petición idéntica a cada DSA subordinado (en ausencia de sombreado). Esto puede hacerse secuencialmente o en paralelo. Típicamente, sólo un DSA será capaz de continuar la resolución de nombres; los otros retornarán el error de servicio **unableToProceed**. En ciertas circunstancias (raras por cierto), es posible que más de un DSA continúen la resolución de nombres, lo que da lugar a resultados duplicados.

#### 8.1.2 Descomposición de petición

La descomposición de la petición misma, que es la otra forma de descomponer una petición, es un proceso realizado internamente por un DSA antes de comunicar con otro u otros DSA. Una petición se descompone en varias subpeticiones, posiblemente diferentes, de modo que cada una de las subpeticiones corresponde a una parte de la tarea original. La descomposición de petición puede utilizarse solamente durante la evaluación de una operación de listar o buscar. Tras la descomposición de la petición, cada una de las subpeticiones puede concatenarse a otros DSA para que continúen la tarea, o se puede retornar un resultado parcial (un referimiento incrustado) al solicitante. Un ejemplo del caso en que se envía la misma subpetición a diferentes DSA se da cuando una inserción tiene referencias y/o NSSR subordinadas que, juntas, referencian más de un DSA. Un ejemplo del caso en que se envían subpeticiones diferentes a los mismos o diferentes DSA se da cuando, durante una operación de buscar (en un subárbol), se encuentran dos inserciones diferentes cada una de las cuales tiene una referencia subordinada.

### 8.2 Uniconcatenación

Este modo de interacción (descrito en la figura 2) puede ser utilizado por un DSA, para pasar una petición a otro DSA, cuando el primero conoce los contextos de denominación que tiene el último. Puede utilizarse la uniconcatenación para entrar en contacto con un único DSA señalado en una referencia de remisión, una referencia subordinada, una referencia superior, una referencia de proveedor o una referencia maestra.

NOTA – En la figura 2, el orden de las interacciones se define por los números asociados a las líneas de interacción.

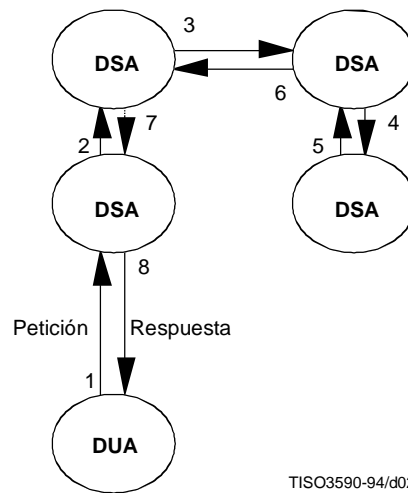


Figura 2 – Modo uniconcatenación

### 8.3 Multiconcatenación

Este modo de interacción es utilizado por un DSA para transferir varias peticiones salientes resultantes de una petición entrante, a resultados de una descomposición de petición o de una descomposición de NSSR.

#### 8.3.1 Multiconcatenación paralela

En la multiconcatenación paralela, el DSA transfiere las peticiones salientes simultáneamente (véase la figura 3a). Si bien la multiconcatenación paralela puede ofrecer un mayor rendimiento, en determinadas circunstancias, por ejemplo en presencia de sombreado, puede dar lugar a la recepción de resultados duplicados.

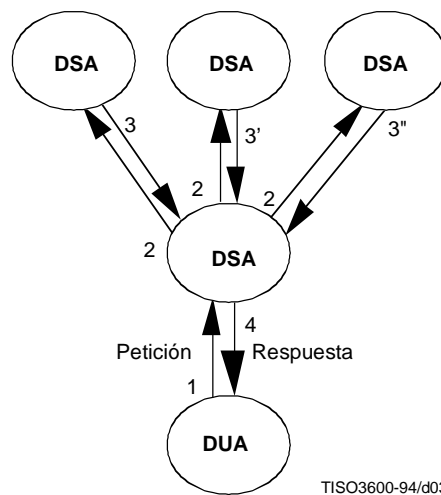
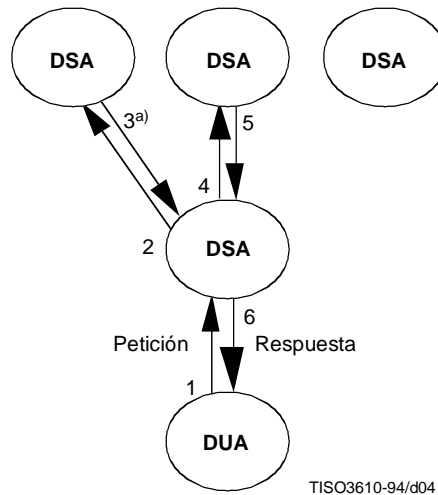


Figura 3a – Multiconcatenación paralela

#### 8.3.2 Multiconcatenación secuencial

En la multiconcatenación secuencial, el DSA transfiere una petición saliente cada vez, y espera el resultado o error de una petición, antes de enviar la siguiente (véase la figura 3b). Si bien la multiconcatenación secuencial puede no ser el modo de interacción más rápido, es improbable que conduzca a la recepción de resultados duplicados.

NOTA – Un DSA puede utilizar una combinación de multiconcatenación paralela y secuencial.



a) Incapaz de avanzar.

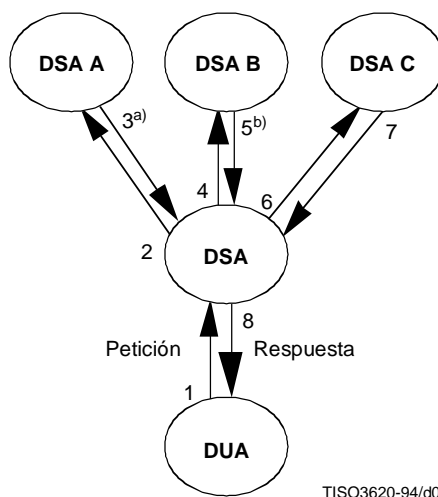
**Figura 3b – Multiconcatenación secuencial (como resultado de la descomposición de una NSSR)**

**8.4 Referimiento**

Un DSA retorna un referimiento (mostrado en las figuras 4a y 4b) en su respuesta a una petición procedente de un DUA o de otro DSA. 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, de remisión, subordinada no específica, de suministrador, o maestra.

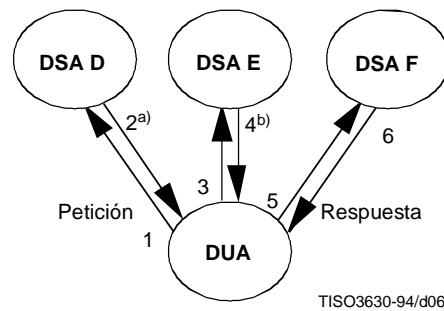
El DSA (figura 4a) que recibe el referimiento puede utilizar la referencia de conocimiento que él contiene para, a continuación, concatenar o difundir (lo cual dependerá del tipo de referencia) la operación original a otros DSA. Como otra posibilidad, un DSA que recibe un referimiento, puede, a su vez, pasarlo en retorno en su respuesta. Un DUA (figura 4b) que reciba un referimiento puede utilizarlo para contactar a otro (u otros) DSA para hacer avanzar la petición.

NOTA – En las figuras 4a y 4b, el orden de interacciones se define por los números asociados a las líneas de interacción.



- a) Referimiento a B.
- b) Referimiento a C.

**Figura 4a – Modo de referimiento (DSA actúa mediante referimientos)**



- a) Referimiento a E.  
 b) Referimiento a F.

**Figura 4b – Modo referimiento  
 (DUA actúa mediante referimientos)**

## 8.5 Determinación del modo

Si un DSA no puede por sí solo resolver una petición, debe concatenar esa petición (o la resultante de su descomposición) a otro DSA, a menos que:

- a) la concatenación ha ya sido prohibida por el usuario mediante los controles de servicio, en cuyo caso el DSA deberá retornar un referimiento o un **chainingRequired serviceError**; o
- b) el DSA tenga razones administrativas, operacionales o técnicas para preferir no concatenar, en cuyo caso el DSA deberá retornar un referimiento.

NOTA 1 – Una "razón técnica" para no concatenar es que el DSA identificado en esa referencia de conocimiento no soporta el DSP.

NOTA 2 – Si el control de servicio **localScope** está fijado, el DSA [o el DMD (dominio de gestión de directorio)] deberán, o bien resolver la petición o retornar un error.

NOTA 3 – Si el usuario prefriere referimientos, deberá fijar **chainingProhibited**.

## SECCIÓN 4 – SERVICIO ABSTRACTO DE DSA

### 9 Visión de conjunto del servicio abstracto de DSA

El servicio abstracto de directorio se describe por completo en la Rec. UIT-T X.511 | ISO/CEI 9594-3. Cuando se proporciona este servicio en un entorno distribuido, según el modelo de la cláusula 7, puede considerarse prestado por medio de un conjunto de los DSA. Esto se ilustra en la figura 1.

Para cada operación definida en el servicio abstracto de directorio se define una operación "concatenada" correspondiente en el servicio abstracto de DSA para uso entre los DSA que cooperan en la realización de esa operación de servicio de directorio. Así, un DSA que recibe una operación leer de un DUA pudiera requerir la asistencia de otro DSA (por ejemplo, un DSA que contuviera la inserción de destino, o una copia del mismo) para satisfacer dicha operación, y por lo tanto enviar a ese DSA una operación leer concatenado.

Los tipos de información intercambiados en el servicio abstracto DSA se definen en la cláusula 10. Las operaciones y errores del servicio abstracto de DSA se definen en las cláusulas 11 a 13.

## 10 Tipos de información

### 10.1 Introducción

Esta cláusula identifica, y en algunos casos define, un cierto número de tipos de información que a continuación se emplean para definir varias de las operaciones del servicio abstracto de DSA. Los tipos de información de que se trata son los comunes a más de una operación, los que probablemente estarán vigentes en el futuro, los que o son lo bastante complejos o autónomos como para merecer que se les defina con independencia de la operación que los utiliza.

## ISO/CEI 9594-4 : 1998 (S)

Varios de los tipos de información empleados en la definición del servicio abstracto de DSA se definen en otro lugar. La subcláusula 10.2 identifica estos tipos e indica la fuente de su definición. Cada una de las subcláusulas restantes (de la 10.3 a la 10.9) identifica y define un tipo de información.

### 10.2 Tipos de información definidos en otro lugar

En la Rec. UIT-T X.501 | ISO/CEI 9594-2 se definen los siguientes tipos de información:

- **aliasedEntryName;**
- **DistinguishedName;**
- **Name;**
- **RelativeDistinguishedName.**

En la Rec. UIT-T X.511 | ISO/CEI 9594-3 se definen los siguientes tipos de información:

(Vincular)

- **DirectoryBind**

(Operaciones)

- **Abandon**

(Errores)

- **abandoned;**
- **attributeError;**
- **nameError;**
- **securityError;**
- **serviceError;**
- **updateError.**

(Clase de objeto de información)

- **OPTIONALLY-PROTECTED**

(Tipo de datos)

- **SecurityParameters**

En la Rec. UIT-T X.520 | ISO/CEI 9594-6 se define el siguiente tipo de información:

- **PresentationAddress.**

### 10.3 Argumentos de concatenación

Los **ChainingArguments** están presentes en toda operación concatenada (chained), para transmitir a un DSA 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 { nameResolutionPhase notStarted },
    traceInformation    [3]  TraceInformation,
    aliasDereferenced   [4]  BOOLEAN DEFAULT FALSE,
    aliasedRDNs         [5]  INTEGER OPTIONAL,
                          -- only present in 1988 systems
    returnCrossRefs     [6]  BOOLEAN DEFAULT FALSE,
    referenceType       [7]  ReferenceType DEFAULT superior,
    info                [8]  DomainInfo OPTIONAL,
    timeLimit           [9]  UTCTime OPTIONAL,
```

<b>securityParameters</b>	[10]	<b>SecurityParameters DEFAULT { }</b> ,
<b>entryOnly</b>	[11]	<b>BOOLEAN DEFAULT FALSE,</b>
<b>uniqueIdentifier</b>	[12]	<b>UniqueIdentifier OPTIONAL,</b>
<b>authenticationLevel</b>	[13]	<b>AuthenticationLevel OPTIONAL,</b>
<b>exclusions</b>	[14]	<b>Exclusions OPTIONAL,</b>
<b>excludeShadows</b>	[15]	<b>BOOLEAN DEFAULT FALSE,</b>
<b>nameResolveOnMaster</b>	[16]	<b>BOOLEAN DEFAULT FALSE,</b>
<b>operationIdentifier</b>	[17]	<b>INTEGER OPTIONAL }</b>

Los diversos componentes tienen los significados que se definen a continuación:

- a) El componente **originador** (*originator*) transporta el nombre del (último) originador de la petición, a menos que haya sido ya especificado en los parámetros de seguridad. Si el **peticionario** (*requester*) está presente en **CommonArguments**, este argumento puede omitirse.

NOTA 1 – Cuando el originador tiene nombres alternativos diferenciados por el contexto, el nombre que se ha de utilizar como valor del **originator** será el nombre distinguido primario, si se conoce. De otro modo, el control de autenticación y acceso basado en el valor del **originator** puede no funcionar como se desea.

- b) El componente **targetObject** lleva el nombre del objeto a cuya inserción de directorio se encamina. El rol de este objeto depende de la operación en cuestión: puede ser el objeto sobre cuya inserción se va a actuar o que va a ser el objeto de base para una petición o subpetición en la que participen múltiples objetos (por ejemplo, **chainedList** o **chainedSearch**). Este componente sólo puede omitirse si tiene el mismo valor que el parámetro de objeto o de objeto de base en la operación concatenada, en cuyo caso su valor implícito será ese valor.

Cuando el **targetObject** incluye los RDN que contienen pares de tipo y valor de atributo para los cuales hay múltiples valores distinguidos diferenciados por el contexto, los RDN que se hayan resuelto serán RDN primarios.

- c) El componente **operationProgress** se utiliza para informar al DSA sobre el avance de la operación, y por ende sobre el papel que se espera que desempeñe en la realización global de la misma. La información transportada en este componente se especifica en 10.5.
- d) El componente **tracelInformation** se emplea para impedir la formación de bucles entre los DSA cuando está actuando la concatenación. Un DSA añade un nuevo elemento a la información de rastreo antes de concatenar una operación a otro DSA. Cuando se pide a un DSA 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 10.6.
- e) 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, una o más inserciones de alias. El valor por defecto de **FALSE** indica que no se ha desreferenciado ninguna inserción de alias.
- f) El componente **aliasedRDNs** indica cuántos de los RDN en el **targetObject Name** se han generado a partir de los atributos **aliasedentryName** de una (o más) inserciones de alias. El valor entero se fija cada vez que una inserción de alias es encontrada y desreferenciada. Este componente estará presente si, y únicamente si, el componente **aliasDereferenced** es **TRUE**.

NOTA 2 – Este componente se proporciona para asegurar la compatibilidad con las implementaciones edición 1988 de directorio. Los DUA (y DSA) implementados con arreglo a ediciones posteriores de las Especificaciones de directorio omitirán siempre este parámetro de lo **CommonArguments** de una petición subsiguiente. De esta manera, el directorio no señalará un error si, como consecuencia de la desreferenciación de alias, se obtienen otros alias.

- g) El componente **entryOnly** se fija a **TRUE** si la operación original fue una búsqueda, fijándose el argumento a **oneLevel** y hallándose una inserción de alias como subordinado inmediato del **baseObject**. El DSA que efectúa con éxito la resolución de nombre sobre el nombre **targetObject** sólo efectuará la evaluación de objeto en la inserción nombrada.
- h) 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 retornadas al DSA inicial como referencias cruzadas, junto con un resultado o un referimiento. El valor por defecto de **FALSE** indica que dichas referencias de conocimientos no deben ser retornadas.

- i) El componente **referenceType** indica, al DSA al que se pide que realice la operación, qué tipo de conocimiento se empleó para encaminar la petición hacia él. El DSA puede así ser capaz de detectar errores en el conocimiento contenido por el invocador. Si se detecta un error, se indicará mediante un **ServiceError** con el problema **invalidReference**. En 10.7 se da una descripción completa de **ReferenceType**.

NOTA 3 – Si el **referenceType** está ausente se supone que el tipo de referencia es **superior**.

- j) El componente **info** se emplea para transportar información específica del DMD entre los DSA que participan en el procesamiento de una petición común. Este componente es del tipo **DomainInfo**, que es un tipo no restringido:

**DomainInfo ::= ABSTRACT-SYNTAX.&Type**

- k) El componente **timeLimit**, si está presente, indica el tiempo transcurrido en el cual la operación debe estar concluida véase 16.1.4.1).
- l) El componente **SecurityParameters** se especifica en la Rec. UIT-T X.511 | ISO/CEI 9594-3. Su ausencia se considera equivalente a que haya un conjunto vacío de parámetros de seguridad.
- m) El componente **authenticationLevel** se suministra opcionalmente cuando se necesita para indicar la forma en que se ha realizado la autenticación. El elemento **authenticationLevel** se describe en la Rec. UIT-T X.501 | ISO/CEI 9594-2.
- n) **uniqueIdentifier** se suministra opcionalmente cuando se necesita para confirmar el nombre del originador. El elemento **uniqueIdentifier** se describe en la Rec. UIT-T X.501 | ISO/CEI 9594-2.
- o) El componente **entryOnly** se fija a **TRUE** si la operación original fue una búsqueda con el argumento **subset** fijado a **oneLevel**, y si se encontró una inserción de alias como subordinado inmediato del **baseObject**. El DSA que efectúa con éxito la resolución en el nombre **targetObject**, efectuará la evaluación de objeto sólo en la inserción nombrada.
- p) El componente **exclusions** sólo es significativo para operaciones de búsqueda; si está presente, indica qué subárboles de inserciones subordinadas al **targetObject** no se incluirán en el resultado de una operación buscar (véase 10.9).
- q) El componente **excludeShadows** sólo tiene significado para operaciones de buscar y listar; indica que la búsqueda se aplicará a inserciones y no a copias de inserción. Este componente opcional puede ser utilizado por un DSA como unidireccional a fin de evitar la recepción de resultados duplicados (véase 20.1).
- r) El componente **nameResolveOnMaster** sólo es significativo durante la resolución de nombre, y sólo es fijado si se ha encontrado una NSSR. Si se ha fijado a **TRUE**, señala que la resolución de nombre subsiguiente, es decir, la concordancia de los RDN restantes del **nextRDNTToBeResolved**, no empleará información de copia de inserción; una resolución subsiguiente de cada RDN restante deberá hacerse en el DSA maestro para la inserción identificada por ese RDN (véase 20.1).
- s) El componente **operationIdentifier** facilita la correlación de las operaciones DAP con subsiguientes operaciones DSP conexas, así como con los resultados. Este componente es asignado por el DSA, que primero recibe una petición DAP o la copia de los argumentos de concatenación de las peticiones DSP que requieren ulterior concatenación. El DSA que asigna el **operationIdentifier** no volverá a utilizar el entero asignado durante un periodo de tiempo suficientemente prolongado. El DSA facilita la correlación de las peticiones DAP y DSP conexas y sus resultados mediante el registro del **operationIdentifier** para cada operación y resultado, junto con el nombre del DSA que lo asignó (el primer DSA de **traceInformation** en una petición concatenada). Esa correlación debe ser útil a los efectos de la inclusión en el fichero registro cronológico, la auditoría, la tasación y liquidación, etc.

## 10.4 Resultados de concatenación

Los **ChainingResults** están presentes en el resultado de cada operación y proporcionan una retroalimentación al DSA que la invocó.

```
ChainingResults ::= SET {
    info                [0]  DomainInfo OPTIONAL,
    crossReferences     [1]  SEQUENCE OF CrossReference OPTIONAL,
    securityParameters [2]  SecurityParameters DEFAULT { },
    alreadySearched     [3]  Exclusions OPTIONAL }
```

Los distintos componentes tienen los significados que se definen a continuación:

- El componente **info** se emplea para transportar información específica del DMD entre los DSA que participan en el procesamiento de una petición común. Este componente es del tipo **DomainInfo**, que es un tipo no restringido.
- El componente **crossReferences** no está presente en el componente **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 **accessPoint** (véase 10.8).

```
CrossReference ::= SET {
  contextPrefix    [0] DistinguishedName,
  accessPoint     [1] AccessPointInformation }
```

Un DSA puede añadir una **CrossReference** cuando hace concordar una parte del argumento **targetObject** de una operación con uno de sus prefijos de contexto. Es posible que la autoridad administrativa de un DSA siga la política de no retornar ese conocimiento, por lo que, en este caso, no añadirá un ítem a la secuencia.

- El componente **SecurityParameters** se especifica en la Rec. UIT-T X.511 | ISO/CEI 9594-3. Su ausencia se considera equivalente a que haya un conjunto vacío de **parámetros de seguridad**.
- El componente **alreadySearched**, si existe, indica cuál de los RDN subordinados, subordinados a **targetObject** se han procesado como parte de una operación buscar concatenado y por tanto no serán incluidas en una subpetición posterior.

NOTA – Los nombres en **contextPrefix** o **alreadySearched** serán nombres distinguidos primarios y no contendrán nombres distinguidos alternativos.

## 10.5 Avance (o progreso) de la operación

El valor **OperationProgress** describe el grado de avance en la ejecución de una operación en la que deben participar varios DSA.

```
OperationProgress ::= SET {
  nameResolutionPhase    [0] ENUMERATED {
                                notStarted      (1),
                                proceeding       (2),
                                completed       (3) },
  nextRDNTToBeResolved  [1] INTEGER OPTIONAL }
```

Los distintos componentes tienen los significados que se definen a continuación:

- 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 DSA con un contexto de denominación que contenga el o los RDN iniciales del nombre. Si la resolución de nombre está **proceeding**, se ha reconocido la parte inicial del nombre, si bien todavía no se ha alcanzado el DSA que contiene el objeto de destino. El componente **nextRDNTToBeResolved** indica cuánto ha sido reconocido del nombre [véase 10.5 b)]. Si la resolución de nombre está **completed**, se ha alcanzado el DSA que contiene el objeto de destino y la ejecución de la operación propiamente dicha está en curso.
- El componente **nextRDNTToBeResolved** indica al DSA cuál de los RDN 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 RDN en el nombre. Este componente sólo está presente si el componente **nameResolutionPhase** tiene el valor **proceeding**.

## 10.6 Información de rastreo

El valor **TracelInformation** contiene una anotación de los DSA que han participado en la ejecución de una operación. Se utiliza para detectar, o evitar, la existencia de bucles que podrían producirse debido a un conocimiento imperfecto o a la presencia de bucles de alias en el DIT.

```
TracelInformation ::= SEQUENCE OF Traceltem
```

```
Traceltem ::= SET {
  dsa                [0] Name,
  targetObject       [1] Name OPTIONAL,
  operationProgress  [2] OperationProgress }
```



Todo DSA que propaga una operación a otro DSA añade un nuevo ítem al final de la secuencia de **Traceltem**. Cada uno de esos **Traceltem** contiene:

- a) El nombre del dsa que agrega el ítem.
- b) El nombre **targetObject** que fue recibido, en la petición entrante, por el DSA que añade el ítem. Este parámetro se omite si la petición que se está concatenando procede de un DUA (en cuyo caso el valor implícito es el **object** o **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 DSA que está añadiendo el ítem.

**dsa** será el nombre distinguido primario y no contendrá nombres distinguidos alternativos. Cada RDN en **targetObject** que haya sido procesado será un RDN primario. En el componente **valuesWithContext** de **AttributeTypeAndDistinguishedValue** en el RDN pueden incluirse valores distinguidos alternativos con contextos.

## 10.7 Tipo de referencia

Un valor **ReferenceType** indica una de las diversas clases de referencias que se definen en la Rec. UIT-T X.501 | ISO/CEI 9594-2.

```
ReferenceType ::= ENUMERATED {
    superior          (1),
    subordinate       (2),
    cross             (3),
    nonSpecificSubordinate (4),
    supplier         (5),
    master           (6),
    immediateSuperior (7),
    self             (8) }
```

## 10.8 Información de punto de acceso

Hay tres tipos de punto de acceso:

- a) Un valor **AccessPoint** identifica un determinado punto en el que puede haber acceso al directorio, y más concretamente a un DSA. El punto de acceso tiene un **Name**, el del DSA afectado, y una **PresentationAddress**, para uso en comunicaciones OSI a ese DSA.

```
AccessPoint ::= SET {
    ae-title          [0] Name,
    address           [1] PresentationAddress,
    protocollInformation [2] SET OF ProtocolInformation OPTIONAL }
```

- b) Un valor **MasterOrShadowAccessPoint** identifica un punto de acceso al directorio. La **category**, **master** o **shadow**, del punto de acceso depende de si apunta a un contexto de denominación o a un área replicada utilizable en común.

```
MasterOrShadowAccessPoint ::= SET {
    COMPONENTS OF AccessPoint,
    category       [3] ENUMERATED {
        master      (0),
        shadow      (1) } DEFAULT master }
```

- c) Un valor **MasterAndShadowAccessPoints** identifica un conjunto de puntos de acceso al directorio, es decir un grupo de DSA relacionados. Estos puntos de acceso tienen una misma propiedad: cada uno de ellos refiere a un DSA que contiene información de inserción procedente de un contexto de denominación común o un conjunto común de contextos de denominación gobernados por un DSA cuando el valor es un valor del atributo **nonSpecificKnowledge**. Un valor **MasterAndShadowAccessPoints** indica la **category** de cada valor de **AccessPoint** que contiene. El punto de acceso del DSA maestro del contexto de denominación no tiene que estar incluido en el conjunto.

```
MasterAndShadowAccessPoints ::= SET OF MasterOrShadowAccessPoint
```

Un valor **AccessPointInformation** identifica uno o más puntos de acceso al directorio.

```
AccessPointInformation ::= SET {
    COMPONENTS OF MasterOrShadowAccessPoint,
    additionalPoints [4] SET OF MasterOrShadowAccessPoint OPTIONAL }
```

En el caso de los DSA edición 1988 que producen un valor **AccessPointInformation**, está ausente el componente opcional del conjunto. En el caso de los DSA edición 1988 que interpretan un valor **AccessPointInformation**, se ignoran cualesquiera valores **MasterAndShadowAccessPoints** presentes.

En el caso de los DSA edición 1998, el componente de valor **MasterOrShadowAccessPoint** producido para un valor **AccessPointInformation** puede ser de la categoría maestro o sombra, en función del procedimiento de selección de conocimiento del DSA que produce el valor. Puede considerarse como un punto de acceso sugerido, proporcionado por el DSA que genera el valor para el DSA que lo recibe. Opcionalmente puede también producirse un conjunto de valores **MasterAndShadowAccessPoints** para un valor **AccessPointInformation**. Esto constituye información adicional que puede utilizarse en el procedimiento de selección del conocimiento del DSA receptor para determinar un punto de acceso alternativo.

## 10.9 Exclusiones

Como se define en 10.3, el componente **exclusions** de **ChainingArguments** se utiliza para limitar el alcance de una operación de buscar identificando un número de inserciones subordinados al objeto pretendido los cuales no serán incluidos en el procesamiento de una operación de buscar, así como tampoco todos sus subordinados. El componente **exclusion** se define como un valor del tipo ASN.1 **Exclusions**.

**Exclusions ::= SET OF RDNSequence**

Cada valor **RDNSequence** en el conjunto **Exclusions** debe identificar el prefijo de contexto de un contexto de denominación subordinado al objeto pretendido. Si un DSA recibe una petición de **buscar** o un valor **RDNSequence** que no es conforme con esta constricción, el DSA puede ignorar dicho valor. La **RDNSequence** es relativa al objeto pretendido, y no es el nombre distinguido del prefijo de contexto.

Las **Exclusions** serán nombres distinguidos primarios. Asimismo, se pueden incluir nombres distinguidos alternativos e información de contexto.

Además de formar parte de una petición de usuario, **Exclusions** puede ser utilizado por DSA para minimizar la información duplicada retornada como consecuencia de subpeticiones de buscar ejecutadas en presencia de información sombreada.

La figura 5 muestra un ejemplo de la utilización de **Exclusions**. En este ejemplo, un DSA contiene dos áreas replicadas, una debajo de la otra. Una comienza por el prefijo de contexto X, y la otra por el prefijo de contexto C. Una copia de inserción en Y tiene tres referencias subordinadas a contextos de denominación: A, B y C.

Si, a título de ejemplo, se efectúa una búsqueda de subárbol en este DSA, comenzando por un objeto de base con el contexto de denominación X, el DSA puede proporcionar información tomada de áreas replicadas X y C. La información procedente de contextos de denominación A y B debe ser proporcionada vía las referencias subordinadas. Cuando se efectúa la descomposición de una petición, referencias de continuación, que serán utilizadas o bien en **partialResults** o concatenación, especificarán Y como el objeto pretendido y C como un elemento simple de un conjunto **Exclusions**.

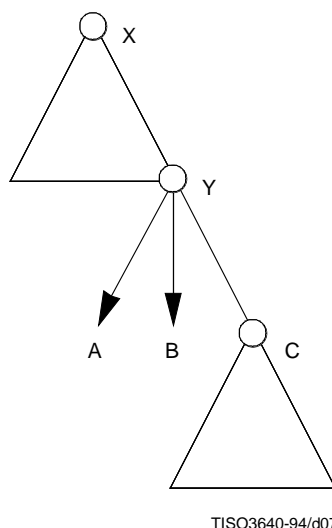


Figura 5 – Exclusions (exclusiones)

## 10.10 Referencia de continuación

Una **ContinuationReference** describe cómo continuar la ejecución de toda o parte de una operación en uno o unos DSA diferentes. Típicamente, se retorna como un referimiento cuando el DSA que interviene no puede o no quiere propagar la petición propiamente dicha.

```
ContinuationReference ::= SET {
    targetObject          [0] Name,
    aliasedRDNs           [1] INTEGER OPTIONAL, -- only present in 1988 systems
    operationProgress     [2] OperationProgress,
    rdnsResolved          [3] INTEGER OPTIONAL,
    referenceType         [4] ReferenceType,
    accessPoints          [5] SET OF AccessPointInformation,
    entryOnly             [6] BOOLEAN DEFAULT FALSE,
    exclusions            [7] Exclusions OPTIONAL,
    returnToDUA           [8] BOOLEAN DEFAULT FALSE,
    nameResolveOnMaster  [9] BOOLEAN DEFAULT FALSE }
```

Los distintos componentes tienen los significados que se definen a continuación:

- a) El **targetObject Name** indica el nombre 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.

Los RDN en **targetObject** serán RDN primarios (para los RDN ya procesados). Se podrían incluir valores distinguidos alternativos con contexto.

- b) El componente **aliasedRDNs** indica cuántos de los RDN (en su caso), del objeto pretendido, han sido producidos desreferenciando un alias. El argumento sólo está presente si se ha desreferenciado un alias.

NOTA – Este componente se proporciona para asegurar la compatibilidad con las implementaciones edición 1988 de directorio. Los DUA (y DSA) implementados con arreglo a ediciones posteriores de las Especificaciones de directorio omitirán siempre este parámetro de los **CommonArguments** de una petición subsiguiente. De esta manera, el directorio no señalará un error si, como consecuencia de la desreferenciación de alias, se obtienen otros alias.

- c) El **operationProgress** indica la cantidad de resolución de nombre que se ha alcanzado, y que regirá la ulterior ejecución de la operación por los DSA denominados, en caso de que el DSA o DUA que reciba la **ContinuationReference** desee seguirla.
- d) El valor del componente **rdnsResolved** (que sólo debe estar presente si algunos de los RDN del nombre no han sido sometidos a resolución de nombre total, pero se ha supuesto que son correctos, como consecuencia de una referencia cruzada) indica cuántos RDN se han resuelto efectivamente utilizando referencias internas únicamente.
- e) El componente **referenceType** indica qué tipo de conocimiento se utilizó para generar esta continuación.
- f) El componente **accessPoints** indica los puntos de acceso que han de contactarse para efectuar esta continuación. Sólo donde intervengan referencias subordinadas no específicas puede haber más de un ítem de **AccessPointInformation**.
- g) El componente **entryOnly** se fija a **TRUE** si la operación original era una búsqueda, con el argumento **subset** fijado a **oneLevel**, y se encontró una inserción de alias como subordinada inmediata del **baseObject**. El DSA que efectúe de manera satisfactoria la resolución de nombre en el nombre del **targetObject**, efectuará la evaluación de objeto solamente en la inserción nominada.
- h) El componente **exclusions** identifica un conjunto de contextos de denominación subordinados que no deben ser explorados por el DSA receptor.
- i) El elemento **returnToDUA** se proporciona facultativamente cuando el DSA que utiliza la referencia de continuación desea indicar que no está dispuesto a retornar información vía un DSA intermedio (por ejemplo, por razones de seguridad) y desea indicar que la información puede estar directamente disponible mediante una operación a través de DAP entre el DUA originador y el DSA. Cuando **returnToDUA** está fijado a **TRUE**, se puede fijar **referenceType** a **self**.
- j) El elemento **nameResolveOnMaster** se suministra facultativamente cuando el DSA que crea la referencia de continuación ha encontrado NSSR. Si está fijado a **TRUE**, este elemento señala que la subsiguiente resolución de nombre, es decir, la concordancia de los restantes RDN de **nextRDNTobeResolved**, no emplearán información de copia de inserción; la resolución subsiguiente de cada RDN restante se efectuará en el DSA maestro para la inserción identificada por ese RDN (véase 20.1).

## 11 Vincular y desvincular

Un DSA utiliza **DSABind** y **DSAUnbind**, respectivamente, al comienzo y al final de un periodo de acceso a otro DSA.

### 11.1 DSA Bind (DSA vincular)

Se utiliza un **DSABind** para iniciar un periodo de cooperación entre dos DSA que proporcionan el servicio de directorio.

**DSABind ::= BIND**

<b>ARGUMENT</b>	<b>DirectoryBindArgument</b>
<b>RESULT</b>	<b>DirectoryBindResult</b>
<b>BIND-ERROR</b>	<b>DirectoryBindError</b>

Los componentes del **DSABind** son idénticos a sus contrapartidas en el **DirectoryBind** (véase la Rec. UIT-T X.511 | ISO/CEI 9594-3) con las siguientes diferencias:

- Las **Credentials** de **DirectoryBindArgument** permiten enviar información identificadora del título-AE del DSA iniciador al DSA respondedor. El título-AE se enviará en la forma de nombre distinguido de directorio.
- Las **Credentials** de **DirectoryBindResult** permiten enviar la información identificadora del título-AE del DSA respondedor al DSA iniciador. El título-AE se enviará en la forma de nombre distinguido.
- El nombre del DSA o título-AE puede utilizar nombres distinguidos alternativos y contener información de contexto.

NOTA 1 – Cuando se utilizan nombres en credenciales simples o fuertes, es posible utilizar nombres distinguidos alternativos si éstos existen. No obstante, si no se utiliza el nombre distinguido primario, el control de autenticación y de acceso basado en el nombre puede no funcionar como se desea. Tras el procesamiento satisfactorio de una operación BIND autenticada, sea cual fuere el nombre utilizado en el argumento BIND, las entidades vinculadas se conocerán entre sí a partir de entonces por sus nombres distinguidos primarios, para facilitar la operación de control de acceso cuando el BIND está en efecto.

NOTA 2 – Las credenciales necesarias para la autenticación pueden ser transportadas por el elemento de servicio de intercambio de seguridad (véase la Rec. UIT-T X.519 | ISO/CEI 9594-5), en cuyo caso éstas no están presentes en los argumentos o resultados de vinculación.

### 11.2 DSA Unbind (DSA desvincular)

Se utiliza un **DSAUnbind** para finalizar un periodo de cooperación entre dos DSA que proporcionan el servicio de directorio.

**DSAUnbind ::= UNBIND**

No hay argumentos, resultados, ni errores.

## 12 Operaciones concatenadas

Para cada una de las operaciones utilizadas para acceder al servicio abstracto de directorio se utiliza una operación entre DSA cooperantes en una correspondencia de uno a uno. Los nombres de las operaciones se han elegido de modo que reflejen esa correspondencia prefijando los nombres de operaciones utilizadas entre DSAs cooperantes con el término "concatenado" ("chained").

Los argumentos, resultados y errores de las operaciones concatenadas se forman sistemáticamente, salvo en un caso, a partir de los argumentos, resultados y errores de las correspondientes operaciones abstractas del servicio abstracto de directorio (tal como se describe en 12.1). La excepción es la operación **ChainedAbandon**, que es sintácticamente equivalente a su contrapartida del servicio de directorio (descrita en 12.2).

### 12.1 Operaciones concatenadas

Un DSA que haya recibido una operación de un DUA puede optar por constituir una forma concatenada de esa operación para propagarla a otro DSA. Un DSA que haya recibido una forma concatenada de una operación también puede, a su vez, optar por concatenarla a otro DSA. El DSA que invoca la forma concatenada de una operación puede firmar, criptar o firmar y criptar el argumento de la operación; si así se solicita, DSA que efectúa la operación puede firmar, criptar, o firmar y criptar el resultado o el error devuelto por el respondedor de la operación.

La forma concatenada de una operación se especifica utilizando el tipo parametrizado **chained { }**.

```

chained { OPERATION : operation } OPERATION ::= {
  ARGUMENT      OPTIONALLY-PROTECTED {
    SET {
      chainedArgument      ChainingArguments,
      argument              [0] operation.&ArgumentType },
      DIRQOP.&dspChainedOp-QOP{@dirqop} }
    RESULT      OPTIONALLY-PROTECTED {
      SET {
        chainedResult      ChainingResults,
        result              [0] operation.&ResultType },
        DIRQOP.&dspChainedOp-QOP{@dirqop} }
      ERRORS { operation.&Errors EXCEPT (referral | dsaReferral) }
      CODE operation.&operationCode }
    ERRORS { operation.&Errors EXCEPT (referral | dsaReferral) }
  CODE operation.&operationCode }

```

NOTA 1 – Las operaciones del servicio abstracto de directorio que pueden utilizarse como el parámetro efectivo de **chained { }** incluyen el error **abandoned**. La presencia de este error en el conjunto de posibles errores de una operación concatenada refleja la posibilidad, discutida en 12.2, de que se pueda generar un **ChainedAbandon** para una operación **ChainedModify** cuando fracasa una asociación ligada.

NOTA 2 – La especificación definitiva del servicio abstracto de DSA en el anexo A aplica este tipo parametrizado para construir todas las operaciones concatenadas del servicio abstracto.

El argumento de la operación derivada tiene los siguientes componentes:

- a) **chainedArgument** – Este es un valor de **ChainingArguments** que contiene esa información, por encima del argumento original suministrado por el DUA, que se necesita para que el DSA actuante efectúe la operación. Este tipo de información se define en 10.3.
- b) **argument** – Este es un valor **operation.&Argument** y está constituido por el argumento original suministrado por el DUA, como se especifica en la cláusula pertinente de la Rec. UIT-T X.511 | ISO/CEI 9594-3.

Si la respuesta tiene éxito, el resultado de la operación derivada consta de los siguientes componentes:

- a) **chainedResult** – Este es un valor de **ChainingResults** que contiene esa información, por encima de la que se suministrará al DUA de origen, la cual podrán necesitar los DSA precedentes en una cadena. Este tipo de información se define en 10.4.
- b) **result** – Este es un valor **operation.&Result** y está constituido por el resultado que retorna el DSA que efectúa esta operación, y que va a ser incluido en el resultado que se devolverá al DUA de origen. Esta información se especifica en la cláusula pertinente de la Rec. UIT-T X.511 | ISO/CEI 9594-3.

Si la petición fracasa, se retorna uno de los errores del conjunto **operation.&Errors**, con la diferencia de que se retorna **dsaReferral** en lugar de **referral**. El conjunto de los errores que pueden señalarse se describe para la operación correspondiente en la Rec. UIT-T X.511 | ISO/CEI 9594-3. El error **dsaReferral** se describe en 13.2.

## 12.2 Operación ChainedAbandon

Un DSA usa una operación **chainedAbandon** para indicar a otro DSA 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:

- la propia operación que llevó originalmente a la concatenación del DSA ha sido abandonada, o ha sido abortada implícitamente por la ruptura de una asociación;
- el DSA ha obtenido la información necesaria de otra manera, por ejemplo, de un DSA de respuesta más rápida que participa en la multiconcatenación paralela.

Un DSA nunca está obligado a emitir un **chainedAbandon**, ni de hecho a abandonar efectivamente una operación cuando se le pide que lo haga.

Si **chainedAbandon** logra efectivamente detener la ejecución de una operación, se retornará un resultado y la operación en cuestión retornará un error **abandoned**. Si **chainedAbandon** no logra detener la operación, ella misma retornará un error de **abandonFailed**.

### 12.3 Operaciones concatenadas y versión de protocolo

Las operaciones que requieren una versión de protocolo mayor que v1 (como la operación **modifyEntry** con ciertos argumentos) o que devuelven diferentes resultados cuando se utilizan con una versión de protocolo superior a v1 (como la operación **modifyEntry** con un argumento firmado) sólo se concatenarán en asociaciones con el número de versión igual o superior al utilizado para transmitir la petición.

## 13 Errores concatenados

### 13.1 Introducción

En su mayor parte, pueden ser retornados en el servicio abstracto de DSA los mismos errores que en el servicio abstracto de directorio. Las excepciones son que se retorna el "error" **dsaReferral** (véase 13.2) en vez de **Referral**, y que los siguientes problemas de servicio tienen la misma sintaxis abstracta pero diferente semántica:

- a) **invalidReference** – El DSA que retorna este error detectó un error en el conocimiento del DSA llamante según lo especificado en el argumento de concatenación del **referenceType**.
- b) **loopDetected** – El DSA que retorna este error detectó un bucle en la información de conocimiento de directorio.

La precedencia de los errores que puedan producirse es la misma que para el servicio abstracto de directorio, como se especifica en la Rec. UIT-T X.511 | ISO/CEI 9594-3.

Si tiene lugar un error durante una operación concatenada, el DSA que responde puede firmar, criptar, o firmar y criptar el error devuelto.

### 13.2 Remisión de DSA

El error **dsaReferral** lo genera un DSA cuando, por la razón que sea, no desea seguir ejecutando una operación concatenándola a otro u otros DSA. Las circunstancias en las que puede retornar una remisión se describen en 8.3. Si los parámetros de la operación fueron firmados por el peticionante, el directorio aplica la protección especificada en DIRQOP.

```
dsaReferral ERROR ::= {
    PARAMETER    OPTIONALLY-PROTECTED {
        SET {
            reference      [0] ContinuationReference,
            contextPrefix  [1] DistinguishedName OPTIONAL,
            COMPONENTS OF CommonResults },
        DIRQOP.&dsaReferral-QOP{@dirqop} }
    CODE         id-errcode-dsaReferral }
```

Los diversos parámetros tienen los significados siguientes:

- a) La **ContinuationReference** contiene la información que necesita el invocador para propagar una petición ulterior adecuada, quizá a otro DSA. Este tipo de información se especifica en 10.10.
- b) Si el componente **returnCrossRefs** de los **ChainingArguments** para esta operación, tenía el valor **TRUE**, y el referimiento se basa en una referencia subordinada o de remisión, se puede incluir facultativamente el parámetro **contextPrefix**. La autoridad administrativa de cualquier DSA decidirá qué referencias de conocimiento pueden retornarse, si es que se retorna alguna, de esta manera (las otras podrían, por ejemplo, ser confidenciales para ese DSA).

Un **contextPrefix** o una referencia de continuación constituirán el nombre distinguido primario. Dentro del componente **valuesWithContext** de un **AttributeTypeAndDistinguishedValue** de cualquier RDN se pueden incluir valores distinguidos alternativos con contexto.

## SECCIÓN 5 – PROCEDIMIENTOS DISTRIBUIDOS

### 14 Introducción

#### 14.1 Alcance y límites

Esta cláusula especifica los procedimientos para operación distribuida de directorio ejecutados por los DSA. Cada DSA ejecuta individualmente los procedimientos descritos más adelante; la acción colectiva de todos los DSA produce todo el conjunto de servicios que el directorio proporciona a los usuarios.

#### 14.2 Conformidad

La descripción de los procedimientos del DSA de esta cláusula se basa en los modelos de las cláusulas 8 y 9 de la Rec. UIT-T X.501 | ISO/CEI 9594-2 y las cláusulas 7 y 8. Los diagramas de flujo y sus correspondientes descripciones textuales constituyen un medio de hacer corresponder un determinado conjunto de entradas externas (DAP y/o DSP) de un DSA a una o más salidas externas (es decir, un resultado, error, referimiento o peticiones concatenadas) producidas por ese DSA, en función del particular árbol de información de DSA contenido por ese DSA.

Es probable que el directorio esté distribuido a través de DSA implementados de acuerdo con diferentes ediciones de las Especificaciones de directorio, por ejemplo las de 1988, 1993 y 1997. El DUA que inicia la petición no sabrá conforme a qué edición de las especificaciones ha sido implementado el DSA, o los DSA que están tratando la petición del DUA. En consecuencia, para permitir la operación en ese entorno heterogéneo, un DSA deberá ser implementado de acuerdo con las reglas de extensibilidad definidas en la cláusula 7 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

Una implementación de DSA será funcionalmente equivalente al comportamiento externo especificado por los procedimientos aquí descritos. Los algoritmos utilizados por una implementación particular de DSA para derivar la(s) salida(s) correcta(s) a partir de las entradas dadas y del árbol de información del DSA contenido no están normalizadas.

NOTA – Los diagramas de flujo que acompañan los procedimientos tienen por objeto servir de ayuda a la comprensión de los procedimientos. No deben ser considerados como una alternativa precisa a las descripciones de texto. Cuando haya disparidad entre la descripción textual y el diagrama de flujo para un procedimiento particular, la descripción textual tendrá precedencia.

##### 14.2.1 Interacción entre DSA edición 1988 y edición 1988/1997

Si las operaciones de modificar evalúan más allá de los confines del DSA [es decir, **addEntry** con **TargetSystem**, **Remove** (suprimir) o **Rename** (redenominar) un prefijo de contexto], esta Especificación de directorio sólo especifica cómo se comportarán dos DSA edición 1997, o cómo se comportará una edición 1993 y una edición 1997. La interacción entre dos DSA edición 1988, o entre un DSA edición 1988 y un DSA edición 1993/1997 está fuera del ámbito de las Especificaciones de directorio. Cuando varios DSA edición mixta tienen una vinculación operacional jerárquica, el conocimiento de la edición de cada uno de los demás permite dar al usuario un error consistente.

#### 14.3 Modelo conceptual

La complejidad de la operación distribuida de directorio plantea la necesidad de una modelación conceptual utilizando técnicas descriptivas, tanto narrativas como pictóricas. No obstante, ni los textos narrativos, ni los diagramas gráficos deberán interpretarse como una descripción formal de la operación distribuida de directorio.

#### 14.4 Operación individual y cooperativa de DSA

El modelo contempla la operación de DSA desde dos perspectivas distintas que, en conjunto, dan una imagen operativa completa del directorio.

- a) **Perspectiva centrada en el DSA** – En esta perspectiva, el conjunto de procedimientos que soporta el directorio se describe desde el punto de vista de un DSA individual. Ello hace posible proporcionar una especificación definitiva de cada procedimiento y tener totalmente en cuenta sus interrelaciones y su estructura de control global. Las cláusulas 16 a 22 describen los procedimientos de DSA según una perspectiva centrada en el DSA.
- b) **Perspectiva centrada en la operación** – La perspectiva centrada en el DSA da una descripción con todos sus detalles, pero hace difícil la comprensión de la estructura de las operaciones individuales, que pueden ser procesadas por múltiples DSA. Por ello, en la cláusula 15 se adopta una perspectiva centrada primordialmente en las operaciones para ir presentando las fases de procesamiento aplicables a cada una.

Para soportar la operación distribuida de directorio, cada DSA debe ejecutar las acciones necesarias para realizar el propósito de cada operación y acciones adicionales para distribuir esa realización a través de múltiples DSA. En la cláusula 15 se analiza la distinción entre estas dos clases de acciones. En las cláusulas 16 a 22 se especifican en detalle ambas clases de acciones.

## 14.5 Acuerdos de cooperación entre los DSA

Todos los DSA que están en una relación de subordinado/superior debido a los contextos de denominación que contienen, mantienen entre sí vinculaciones operacionales jerárquicas y/o jerárquicas no específicas, según los tipos de la referencia de conocimiento contenida en el DSA subordinado.

Las vinculaciones operacionales jerárquicas y las jerárquicas no especificadas pueden ser administradas utilizando los procedimientos de las cláusulas 24 y 25 de esta Recomendación | Norma Internacional, o por otros medios (por ejemplo, por teléfono).

Un DSA que contiene inserciones que están comprendidas en el área administrativa de su DSA superior administrará el subesquema y controlará el acceso a las inserciones como lo requiera la autoridad administrativa. La regulación de inserciones dentro de un área administrativa puede efectuarse como se define en la Rec. UIT-T X.501 | ISO/CEI 9594-2, o mediante un mecanismo local.

## 15 Comportamiento de directorio distribuido

### 15.1 Realización cooperativa de operaciones

Cada DSA dispone de procedimientos que le permiten la realización total de todas las operaciones de directorio. Si un DSA contiene íntegramente la DIB, de hecho, todas las operaciones se realizan dentro de ese DSA. Si la DIB está distribuida entre múltiples DSA, la ejecución de una operación típica se fragmenta, realizándose sólo una porción de esa operación en cada uno de los DSA cooperantes, que pueden ser muchos.

En el entorno distribuido, el DSA típico ve cada operación como un suceso transitorio: la operación es invocada por un DUA o por algún otro DSA; el DSA efectúa el procesamiento del objeto y luego lo dirige a otro DSA para ulterior procesamiento.

Otra posible perspectiva considera el procesamiento total de que ha sido objeto una operación durante su realización por múltiples DSA cooperantes. En esta perspectiva se ponen de manifiesto las fases del procesamiento comunes a todas las operaciones.

### 15.2 Fases del procesamiento de operaciones

Cabe descomponer cada operación de directorio en tres distintas fases:

- a) la *fase de resolución de nombre*, en la que el nombre del objeto sobre cuya inserción se va a efectuar una determinada operación se utiliza para localizar el DSA que contiene la inserción;
- b) la *fase de evaluación*, en la que la operación especificada por una petición de directorio particular (por ejemplo, Read) se lleva a cabo efectivamente;
- c) la *fase de fusión de resultados*, en la que los resultados de una operación especificada se retornan al DUA solicitante. Si se eligió un modo de interacción por concatenación, la fase de fusión de resultados puede comprender varios DSA, cada uno de los cuales concatenó la petición o subpetición original (definida en 15.3.1, Descomposición de petición) a otro DSA durante alguna de las fases precedentes, o ambas.

En el caso de las operaciones Read (leer), Compare (comparar), List (listar), Search (buscar), Modify DN (modificar inserción) y Remove Entry (suprimir inserción), la resolución del nombre tiene lugar sobre el nombre de objeto proporcionado en el argumento de la operación. En el caso de Add Entry (añadir inserción) la inserción de destino de la resolución de nombre es la inserción inmediatamente superior al proporcionado en el argumento de la operación – puede derivarse con facilidad suprimiendo el RDN final del nombre proporcionado en el argumento de la operación. (Esto se consigue mediante el argumento local **m** en el procedimiento **FindDSE** de 18.3.1.)



## ISO/CEI 9594-4 : 1998 (S)

Una operación sobre una inserción particular puede ir dirigida inicialmente a cualquier DSA del directorio. Ese DSA utiliza su conocimiento, quizás en conjunción con otros DSA, para procesar la operación en las tres fases.

### 15.2.1 Fase de resolución de nombre

La fase de resolución de nombre es el proceso de concordar secuencialmente cada RDN en un nombre contemplado, con un arco (o vértice) del DIT, comenzando lógicamente por la raíz y descendiendo en el DIT. Sin embargo, puesto que el DIT está distribuido entre un número arbitrariamente grande de DSA, cada DSA sólo podrá efectuar una fracción del proceso de resolución de nombre. Un DSA dado efectúa su parte del proceso de resolución de nombre recorriendo su árbol de información de DSA local. Este proceso se describe en la cláusula 18 y los diagramas que la acompañan (véanse las figuras 9 a 12). Fundándose en su árbol de información de DSA local, y en la información de conocimiento allí contenida, un DSA está en condiciones de inferir si la resolución puede ser continuada por otro u otros DSA, o si el nombre es erróneo.

Si se incluye la opción **manageDSAIT** en controles de servicio, la fase de resolución de nombre está obligada a funcionar dentro de un árbol de información DSA.

### 15.2.2 Fase de evaluación

Concluida la fase de resolución de nombre, se efectúa la operación (por ejemplo, Read o Search) que de hecho se requiere.

Las operaciones que implican la interrogación de una sola inserción – Read y Compare – pueden llevarse a cabo por entero en el DSA en que se ha localizado esa inserción.

Las operaciones que implican la interrogación de múltiples inserciones – List y Search – necesitan localizar subordinados de la inserción pretendida, que pueden hallarse o no en el mismo DSA. Si no están todos en el mismo DSA, hay que dirigir las operaciones a los DSA especificados en las referencias subordinadas, subordinadas no específicas, de suministrador o maestras (según corresponda) para completar el proceso de evaluación.

Si se incluye la opción **managedDSAIT** en controles de servicio, la fase de evaluación está obligada a funcionar dentro de un árbol de información DSA.

### 15.2.3 Fase de fusión de resultados

Se entra en la fase de fusión de resultados una vez que se dispone de algunos de los resultados de la fase de evaluación.

En los casos en que la operación ha afectado solamente a una inserción, puede retornarse simplemente el resultado de la operación al DUA solicitante. Cuando ha afectado a múltiples inserciones en múltiples DSA, los resultados han de combinarse. Si la protección se efectúa en los resultados, éstos no se combinarán. Los resultados se deben devolver al DUA sin efectuar la fusión.

Las respuestas permitidas, retornadas al solicitante tras la fusión, son las siguientes:

- a) un resultado completo de la operación;
- b) un resultado que no está completo porque algunas partes del DIT permanecen inexploradas (sólo en los casos de List y Search). Ese *resultado parcial* puede contener referencias de continuación para las partes del DIT no exploradas;
- c) un error (un referimiento es un caso especial); y
- d) si el solicitante era un DSA, un **ChainingResult**.

## 15.3 Gestión de operaciones distribuidas

En el argumento de cada operación que se le puede pedir que efectúe a un DSA, se incluye información que indica el avance de la operación a medida que transita por distintos DSA de directorio. Esto hace posible que cada DSA realice el aspecto adecuado del procesamiento requerido y también registrar la terminación de ese aspecto antes de dirigir la operación hacia el exterior, a otros DSA.

En el DSA se incluyen procedimientos adicionales para distribuir físicamente las operaciones y satisfacer otras necesidades que plantee esa distribución.

### 15.3.1 Descomposición de petición

La descomposición de petición es un proceso que realiza internamente un DSA antes de comunicar con otro u otros DSA. Una petición se descompone en varias subpeticiones, de tal modo que cada uno de los otros DSA 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 de base. Tras la descomposición, cada subpetición puede ser entonces uniconcatenada o multiconcatenada a otros DSA que continuarán la tarea.

### 15.3.2 DSA como respondedor de petición

Un DSA que recibe una petición puede verificar el grado de avance (o progreso) de esa petición utilizando el parámetro **operationProgress**. Este parámetro 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 deberá tratar de satisfacer el DSA. Si el DSA no puede satisfacer totalmente la petición, deberá pasarla (por uniconcatenación o multiconcatenación) a otro u otros DSA que puedan ayudarle a cumplimentar la petición, o retornar un referimiento a otro DSA, o terminar la petición con un error.

### 15.3.3 Compleción de operaciones

Todo DSA que ha iniciado una operación o la ha propagado a otro u otros DSA debe tener siempre presente la existencia de esa operación hasta que cada uno de los otros DSA haya retornado un resultado o un error, o haya transcurrido el límite tiempo máximo de la operación. Esta exigencia se aplica a todas las operaciones, modos de propagación y fases de procesamiento. Con ello se garantiza el cierre ordenado de operaciones distribuidas que se han propagado y entrado en el directorio.

## 15.4 Tratamiento de bucles

El DIT puede encontrarse en un estado en el que pueden producirse bucles. Por ejemplo, pueden producirse bucles durante una resolución de nombre cuando, como consecuencia de la desreferenciación de uno o más alias, la resolución retorna a la misma rama del DIT. Otra posible causa de formación de bucles es mediante referencias de conocimiento incorrectamente configuradas.

En el contexto de una operación particular de directorio, se produce un bucle si en cualquier momento la operación retorna a un *estado* anterior, donde el estado se define por los siguientes componentes:

- el nombre del DSA que está procesando la operación en ese momento;
- el nombre del **targetObject** contenido en el argumento de la operación;
- el **operationProgress** contenido en el argumento de la operación y definido en 10.5.

Esto no significa que un DSA particular no pueda procesar varias veces una operación. En cambio, sí significa que el DSA no procesará la misma operación varias veces en el mismo estado.

La formación de bucles se controla utilizando el argumento **tracelInformation** definido en 10.6, que registra la secuencia de estados por los que pasa una operación particular. Para determinar la formación de bucles se definen dos estrategias: la detección de bucle y la evitación de bucle, que se describen en 15.4.1 y 15.4.2 respectivamente.

La detección de bucles es obligatoria y la evitación de bucles es opcional.

### 15.4.1 Detección de bucle

Al recibir una operación de directorio, un DSA debe validar inicialmente la operación para asegurarse de que puede hacerse avanzar. Una importante tarea de la validación consiste en comprobar la existencia de bucles, determinando si el estado actual de la operación aparece en la secuencia de estados anteriores registrados en el argumento **tracelInformation** para esa operación. Este paso de la comprobación de bucles es la detección de bucles.

### 15.4.2 Evitación de bucles

La evitación de bucles requiere que un DSA, inmediatamente antes de transferir una operación a otro DSA como parte de un procedimiento de concatenación, determine si el estado consecuente de la operación (que es el **tracelItem** que añadirá el DSA receptor a la **tracelInformation** cuando la reciba) aparece en la secuencia de los estados anteriores registrados en el argumento **tracelInformation** para la operación entrante original.

En el caso en que se reciban o de que se actúe sobre referimientos, la evitación de bucles y la detección de bucles no pueden conseguirse examinando simplemente **tracelInformation**. En este caso, cada vez que un DSA actúa sobre un referimiento, tiene que almacenar el estado consecuente de la operación (es decir, el **tracelItem** que el DSA receptor va a

añadir cuando reciba la petición) junto con una anotación de la petición entrante. Antes de actuar sobre un referimiento o de retornarlo, un DSA tiene que comprobar esta lista, para verificar que anteriormente no se ha enviado una petición idéntica mientras se intentaba atender a la operación entrante.

## 15.5 Otras consideraciones relativas a la operación distribuida

### 15.5.1 Controles de servicio

Algunos controles de servicio requieren una consideración especial en el entorno distribuido para que la operación sea procesada del modo que se solicitó.

- a) **chainingProhibited** – Un DSA consulta este control de servicio para determinar el modo de propagación de una operación. Si está fijado, el DSA utiliza siempre el modo referimiento. Por el contrario, si no lo está, el DSA puede elegir entre emplear concatenación o referimiento según sus capacidades.
- b) **timeLimit** – Un DSA debe tener en cuenta este control de servicio para asegurarse de que no se rebasa el límite de tiempo. El DSA al que un DUA pide que realice una operación toma nota al principio del **timeLimit**, expresado por el DUA como tiempo disponible transcurrido, en segundos, para la ejecución de la operación. Si se requiere concatenación, se incluye el **timeLimit** en el argumento de concatenación que se pasa al (o a los) DSA siguiente(s). 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 restricción especificada originalmente. Si recibe **ChainingArguments** con un **timeLimit** especificado, el DSA receptor respeta ese límite.
- c) **sizeLimit** – Un DSA ha de respetar este control de servicio para garantizar que la lista de resultados no rebasa el tamaño especificado. El límite, tal como figura en el argumento común de la petición original, se transporta inalterado cuando se concatena la petición. Si se requiere una descomposición de la petición, se incluye el mismo valor en el argumento que se pasa al siguiente DSA, esto es, se emplea el límite total para cada subpetición. Cuando se retornan los resultados, el DSA solicitante resuelve los resultados múltiples y aplica el límite al total para asegurar que sólo se retorna el número pedido. Si se ha rebasado el límite, esto se indica en la respuesta.
- d) **priority** – Cualquiera que sea el modo de propagación, cada DSA tiene la responsabilidad de garantizar que en el procesamiento 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 ningún DSA propagará la operación fuera de éste.
- f) **scopeOfReferral** – Si el DSA retorna un referimiento o un resultado parcial a una operación List o Search, las referencias de continuación incrustadas estarán dentro del ámbito solicitado.

Deben respetarse todos los demás controles de servicio, pero su empleo no exige consideración especial alguna en el entorno distribuido.

### 15.5.2 Extensiones

Si un DSA encuentra una operación extendida en la misma fase resolución de nombre del procesamiento y determina que la operación debe concatenarse a uno o más DSA, deberá incluir, sin modificarlas, en la operación abstracta concatenada, todas las extensiones presentes.

NOTA – Una autoridad administrativa puede determinar que es apropiado retornar un **serviceError** con el problema **unwillingToPerform** si no desea propagar una extensión.

Si un DSA encuentra una extensión en la fase de ejecución del procesamiento, se dan dos casos posibles. Si la extensión no es crítica, el DSA la ignorará. Si la extensión es crítica, el DSA retornará un **serviceError** con el problema **unavailableCriticalExtension**. Una extensión crítica a una operación de objeto múltiple puede dar lugar a resultados y errores de servicio de esta variedad. Un DSA que fusiona tales resultados y errores descartará estos errores de servicio y empleará el componente **unavailableCriticalExtension** del **PartialOutcomeQualifier** descrito en la Rec. UIT-T X.511 | ISO/CEI 9594-3.

### 15.5.3 Desreferenciación de alias

La desreferenciación de alias es el proceso de crear un nuevo nombre de objeto pretendido, mediante la sustitución de la parte nombre distinguido de inserción de alias del nombre de objeto pretendido original por el valor del atributo **AliasedEntryName** de dicha inserción de alias. En la operación, el nombre de **object** no es afectado por la desreferenciación de alias.

#### 15.5.4 Resolución de nombres que varían con el contexto

Durante la fase de resolución de nombre, a medida que se procesan los RDN, se crea un nuevo nombre de objeto de destino para garantizar que todos los **AttributeTypeAndDistinguishedValue** en el RDN utilizan el valor distinguido primario de ese atributo como su **value**. De este modo, el nombre de objeto de destino progresa hacia un nombre distinguido primario. Esto se hace para proporcionar un tratamiento de nombre coherente, sobre todo cuando en la resolución de nombre participan DSA anteriores a 1997. El nombre **object** en la operación no se ve afectado por esta sustitución.

#### 15.5.5 DirQOP y petición de protección

El **dirqop** seleccionado en la vinculación DSP y el campo **ProtectionRequest** en la petición DAP gobiernan el funcionamiento de las diversas características de seguridad asociadas con la operación de directorio distribuida.

La selección de protección punto a punto para las peticiones y respuestas DSP (incluidos los argumentos y resultados de concatenación) se indican mediante el **dirqop** establecido entre los DSA en la vinculación DSP.

La selección de la protección preferida por el solicitante para los resultados o errores DAP concatenados se indica mediante el campo **SecurityParameters.ProtectionRequest** en la petición DAP o, si este campo no está presente, el campo **SecurityParameters.ProtectionRequest** en los **ChainingArguments** se ha de fijar para reflejar el **dirqop** en la vinculación DAP.

Cuando se firma una petición DAP, la protección se mantiene en la petición DAP concatenada. Antes de la concatenación se suprime cualquier criptación en una petición DAP, aunque esta información puede protegerse utilizando protección punto a punto de la petición DSP, según las necesidades. La criptación de errores también se aplica únicamente sobre una base punto a punto.

Los parámetros de seguridad gobiernan la operación distribuida de diversas características de seguridad asociadas con una operación de directorio distribuida.

### 15.6 Autenticación de operaciones distribuidas

Usuarios de directorio junto con autoridades administrativas que proporcionan servicios de directorio pueden, si lo desean, exigir que se autentifiquen operaciones de directorio. Para cualquier operación de directorio 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 diversas exigencias de autenticación. Uno de esos conjuntos está constituido por los procedimientos vincular, que facilitan la autenticación entre dos entidades de aplicación de directorio 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 como la proporcionada por vincular, se han definido en el directorio procedimientos adicionales que permiten autenticar operaciones individuales. Están definidos dos conjuntos distintos de procedimientos de autenticación de directorio. Uno de ellos facilita servicios de autenticación de originador, que tratan de la autenticación, por un DSA, 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 retorne.

Para la autenticación de originador se han definido dos procedimientos, uno basado en un simple intercambio de identidades, denominado **autenticación basada en identidad**, y otro basado en técnicas de firma digital, denominado **autenticación basada en firma**. El primero de estos procedimientos es rudimentario ya que el intercambio de identidad se basa en el intercambio de nombres distinguidos que se transmiten en lenguaje ordinario.

Para la autenticación de resultados se ha definido un único procedimiento de **autenticación de resultados**, basado en técnicas de firma digital; debido al carácter generalmente complejo de la colación de resultados, no se ha definido para su autenticación un procedimiento más simple, basado en la identidad.

La autenticación de respuestas de error puede ser soportada por estos procedimientos.

Debe considerarse que los servicios antes descritos aumentan los proporcionados por el servicio vincular. Se supone que los procedimientos vincular se han seguido con éxito antes de la autenticación de operaciones de directorio.

Los procedimientos que debe continuar un DSA al proporcionar autenticación de originador y de resultados se especifican en la cláusula 22.

16 El despachador de operaciones

El **despachador de operaciones** es el principal procedimiento de control en un DSA. El **despachador de operaciones** guía cada operación a través de las tres fases del procesamiento de una petición. Por ello hace uso de un conjunto de procedimientos para procesar la petición por completo, según se muestra en la figura 6.

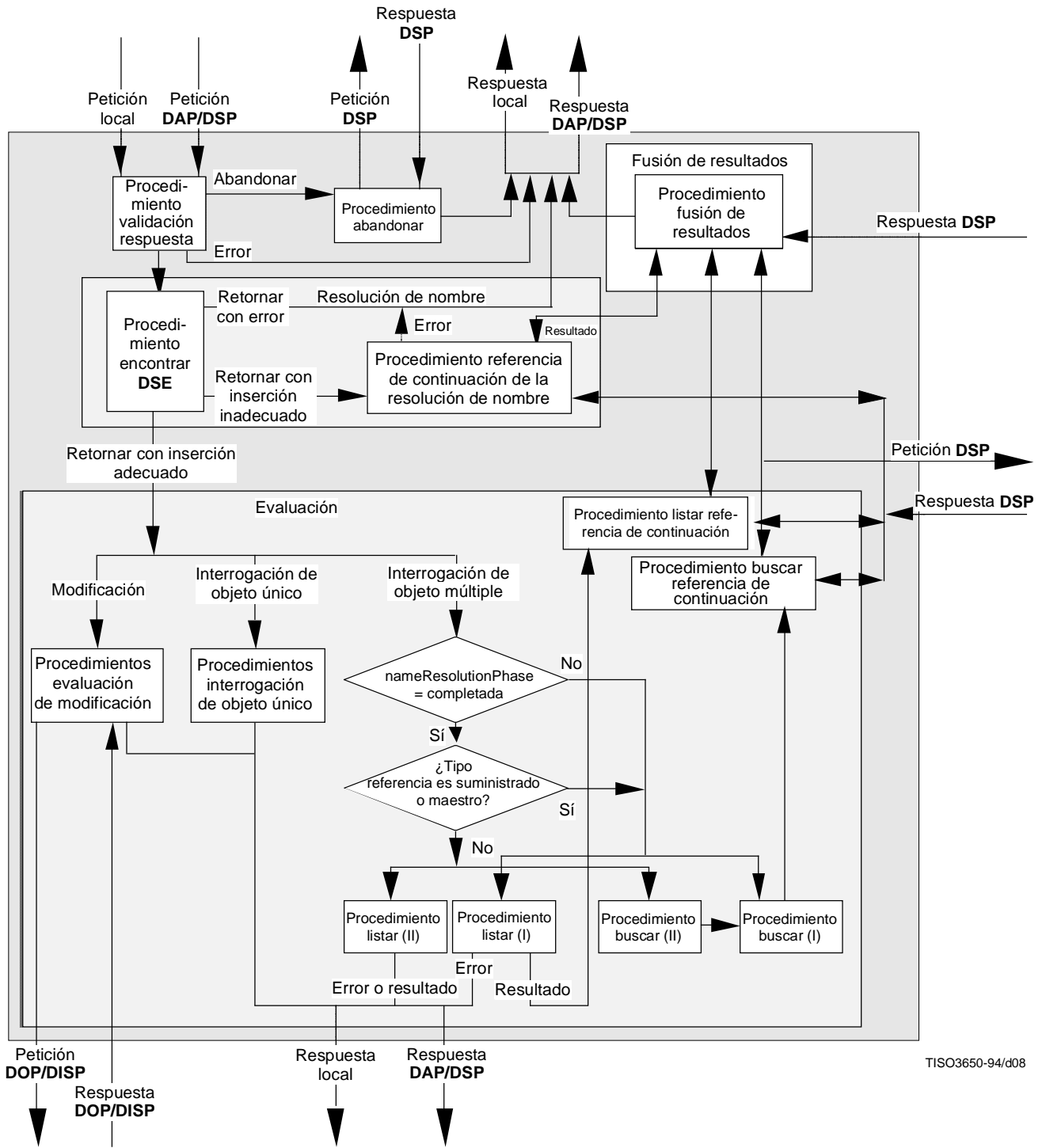
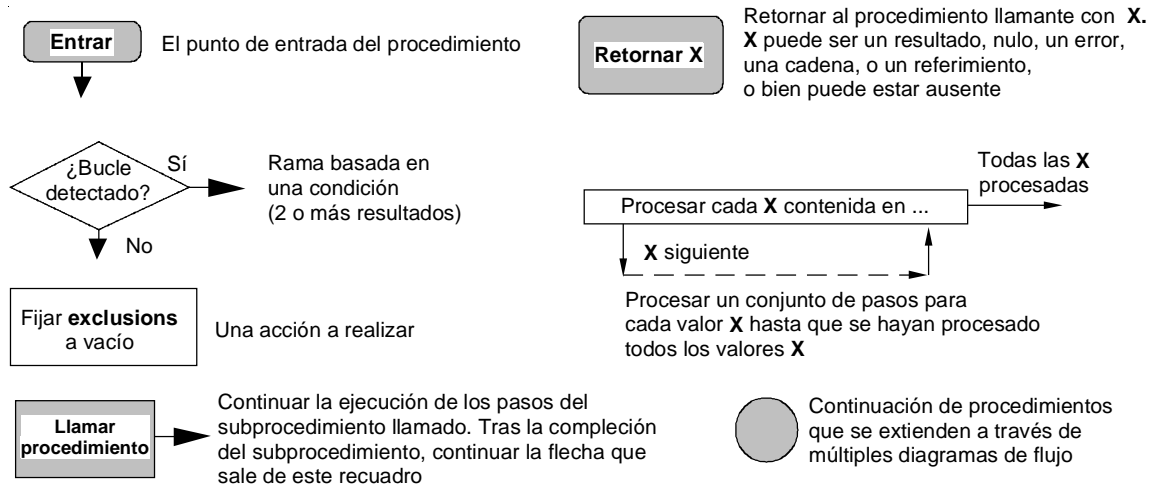


Figura 6 – Despachador de operaciones

## 16.1 Conceptos generales

### 16.1.1 Procedimientos

Cada uno de los procedimientos utilizados por el despachador de operación se compone de una definición de su interfaz conceptual en términos de argumentos, resultados y errores, y de una descripción de los pasos del procedimiento. El comportamiento de los procedimientos se describe mediante diagramas de flujo y texto. En un diagrama de flujo los símbolos utilizados tienen los significados siguientes (véase la figura 7).



TISO3660-94/d09

Figura 7 – Símbolos utilizados en los diagramas de flujo

### 16.1.2 Uso de estructuras de datos comunes

Todos los procedimientos hacen uso de algunas estructuras de datos que están disponibles durante el procesamiento de una operación en el **despachador de operaciones**. Estas estructuras de datos sirven para coordinar el flujo de datos en el **despachador de operaciones**. La mayoría de estas estructuras están asociadas directamente con el argumento de la operación y el resultado a crear para la operación. Se hace referencia a los componentes del argumento y resultado utilizando los nombres que ellos tienen en la definición ASN.1 asociada (por ejemplo, el componente **operationProgress** de los argumentos de concatenación). Si cualquiera de estas estructuras es una estructura compuesta, puede hacerse referencia a un componente de esta estructura como **compound.component** (por ejemplo, **operationProgress.nameResolutionPhase**).

Se definen las siguientes estructuras de datos en el **despachador de operaciones**:

- **NRcontinuationList** – Una lista de referencias de continuación creada para su uso en el procedimiento **Name Resolution Continuation Reference**.
- **SRcontinuationList** – Una lista de referencias de continuación creada para uso en el procedimiento **List o Search Continuation Reference**.
- **admPoints** – Una lista de referencias a DSE de tipo punto administrativo que se forma durante Resolución de Nombre.
- **referralRequests** – Una lista de peticiones o subpeticiones que han sido concatenadas como resultado de referimientos de ejecución. Cada una de estas peticiones/subpeticiones se resume en la forma de un **Traceltem**. Esta lista se utiliza por el procedimiento de evitación de bucles que figura en 15.4.2.

Además, un procedimiento puede utilizar un conjunto de variables definidas localmente.

### 16.1.3 Errores

En toda etapa del procedimiento puede detectarse un error durante la ejecución de cualquier subprocedimiento. El error identificado en este subprocedimiento se retorna normalmente al solicitante como un correspondiente error de protocolo. En este caso es terminado inmediatamente el despachador de operaciones. En caso de que se haya recibido múltiples errores, los procedimientos locales pueden seleccionar uno de ellos para ser retornado.

Como otra posibilidad, un procedimiento puede optar por procesar errores (por ejemplo, si se retorna **serviceError busy** a una subpetición de búsqueda concatenada) en determinados puntos del procesamiento de la operación. En este caso el procedimiento prosigue su ejecución y no se retorna error al solicitante.

Facultativamente, el DSA puede firmar, criptar, o firmar y criptar los errores retornados en una operación distribuida, sobre la base del DirQOP seleccionado y la protección contra error solicitada.

#### 16.1.4 Sucesos asíncronos

Durante el procesamiento de una petición de operación en el despachador de operaciones pueden tener lugar varios acontecimientos asíncronos. Las siguientes subcláusulas especifican cómo manejar un límite de tiempo, un límite de tamaño, o un límite administrativo rebasados, una pérdida de asociación y una petición de abandonar de una operación que se está procesando. El manejo de todos los otros sucesos asíncronos, por ejemplo, las decisiones adoptadas a nivel local, etc., está fuera del alcance de esta Especificación.

##### 16.1.4.1 Límite de tiempo

Un **timeLimit** especificado en los **CommonArguments** puede expirar en cualquier momento de la operación. En este caso, normalmente se retorna un **serviceError** con problema **timeLimitExceeded** al DUA o DSA solicitante y el despachador de operaciones es terminado. Como otra posibilidad, un procedimiento puede optar por manejar este suceso de forma diferente (por ejemplo, durante el procesamiento de una petición de **search**).

Si un DSA recibe una petición de otro DSA con el límite de tiempo excedido, enviará un **serviceError** con un valor de error **timeLimitExceeded** y no continuará el procesamiento de la petición.

Si, cuando expira el **timeLimit**, un DSA tiene (sub)peticiones pendientes, y no hay resultados disponibles, retornará al solicitante un **ServiceError** con un valor de error **timeLimitExceeded**.

Si, cuando expira el límite de tiempo, un DSA tiene subpeticiones pendientes y hay resultados disponibles, retornará al solicitante un resultado con el siguiente contenido:

- a) todos los resultados obtenidos hasta la expiración del **timeLimit**;
- b) el componente **limitProblem** del parámetro de resultado **partialOutcomeQualifier** se fijará a **timeLimitExceeded**;
- c) el componente **unexplored** del parámetro de resultado **partialOutcomeQualifier** contendrá un valor de referencia de continuación para cada uno de los DSA a los que se enviaron subpeticiones pero cuyo resultado no está incluido en el resultado remitido al solicitante, además de referencias de continuación a los DSA a los que este DSA no intentó enviar subpeticiones.

##### 16.1.4.2 Pérdida de una asociación

Si se pierde la asociación con el solicitante, el DSA puede, facultativamente, enviar para cada (sub)petición de interrogación pendiente una petición **chainedAbandon**, a menos que se haya perdido también la asociación con el DSA en cuestión. Todas las respuestas a estas **chainedAbandon** y todas las respuestas a (sub)peticiones pendientes serán descartadas.

Si se pierde la asociación con una de las subpeticiones concatenadas pendientes y no se pierde la asociación con el solicitante, el DSA puede, facultativamente, intentar cualquier referencia alternativa a otro DSA que sea capaz de procesar la petición concatenada (por ejemplo una referencia a un DSA de sombra, tras la pérdida de la asociación con el DSA maestro). Si esto fracasa, el DSA actuará del siguiente modo:

- 1) Si la fase **operationProgress.nameResolution** está fijada a **notStarted** o **proceeding**, retornar o un **serviceError** con problema **unavailable** al solicitante o un error de referimiento cuya referencia de continuación contenga el conjunto de DSA capaces de continuar la operación. Si se utilizan referencias subordinadas no específicas durante la fase de resolución de nombre y no se han perdido todas las asociaciones en cuestión, opcionalmente intentar realizar la resolución del nombre sin los DSA con los que se han perdido las asociaciones. Si esto fracasa, retornar o un **serviceError** con problema **unavailable**, o un error de referimiento que contenga el conjunto completo de NSSR.

NOTA – El tipo de error que se retorna se decidirá a nivel local.

- 2) Si la fase **operationProgress.nameResolution** está fijada a **completed** y la petición es una operación de objeto único, retornar al solicitante un **serviceError** con un parámetro de error **unavailable**.

- 3) Si la fase **operationProgress.nameResolution** está fijada a **completed** y la petición es una operación de interrogación de múltiples inserciones, el DSA añadirá una referencia de continuación al **partialOutcomeQualifier.unexplored** del resultado de la operación, identificando **AccessPointInformation** el conjunto de DSA capaces de proseguir la operación, incluidos los DSA con los cuales se hayan perdido las asociaciones.

#### 16.1.4.3 Abandono de la operación

Durante el procesamiento de una operación puede recibirse una petición de abandonar en cuanto a esa operación. En este caso, durante el procesamiento de la petición de abandonar, se llama al procedimiento **Abandon** para que se encargue de abandonar la operación.

#### 16.1.4.4 Límites administrativos

El administrador del DSA local puede imponer límites administrativos, por ejemplo el periodo de tiempo que habrá de disponerse para el procesamiento de una petición, o el tamaño máximo de los datos que habrán de retornarse. Si se rebasa cualesquiera de estos límites, el DSA retornará **serviceError** con el problema **administrativeLimitExceeded**.

#### 16.1.4.5 Límite de tamaño

Un límite de tamaño, tal como se especifica en **Common Arguments** puede ser rebasado en cualquier punto de tiempo durante el procesamiento de una operación de listar o buscar. En tal caso se retornará al solicitante un resultado parcial (tomado del conjunto de resultados ya obtenidos) con **limitProblem** fijado a **sizeLimitExceeded**. Además, puede utilizarse el componente **unexplored** para retornar referencias de continuación de DSA no accedidos. Después de esto, se termina el despachador de operación.

### 16.2 Procedimientos del despachador de operaciones

El procedimiento que realiza el **despachador de operaciones** para procesar cada petición recibida (bien mediante DAP o DSP) se define por los siguientes pasos. Debido al desreferenciamiento de alias, este procedimiento puede también llamarse a sí mismo (una petición local), en cuyo caso se devuelve una respuesta local (en vez de una respuesta DAP o DSP).

- 1) Validar varios aspectos de los argumentos de la operación (procedimiento **Request Validation**). Si se encuentra un error durante la validación, retornar este error localmente o mediante DAP/DSP.
- 2) Si la operación recibida fue una operación de abandonar, llamar al procedimiento **Abandon** y retornar posteriormente una respuesta.
- 3) Resolver el nombre del objeto de destino ejecutando el procedimiento **Find DSE** (que incluye los subprocedimientos **Target Found** y **Target Not Found**). Si la inserción solicitada fue hallada idónea (de acuerdo con los valores de los controles de servicio, los argumentos de concatenación y las decisiones adoptadas a nivel local), continuar con la **Evaluation Phase** en el paso 6). Si se encontró un error durante Name Resolution, se retorna dicho error. Si se encontró no idónea la inserción, seguir en el paso 4).
- 4) El **Name Resolution Continuation Reference** procedure se llama para procesar la lista de referencias de continuación tal como está almacenada en la **NRcontinuationList**. Para procesar estas referencias de continuación pueden formularse peticiones concatenadas a otros DSA (si los controles de servicio y las decisiones adoptadas a nivel local lo permiten).

En caso de error, este error se retorna directamente bien localmente o mediante DAP/DSP. Si la petición concatenada generó un resultado, continuar en el paso 5).

- 5) Se llama el procedimiento **Result Merging (fusión de resultados)** para fusionar los resultados locales con los resultados concatenados recibidos. Si los resultados concatenados contienen referencias de continuación incrustadas, será posible resolver previamente estas referencias si ello está permitido o prescrito por los controles de servicio y la política local.

Esto puede dar lugar a que se emitan peticiones concatenadas adicionales (cuyos resultados pueden también contener referencias de continuación incrustadas).

Los resultados fusionados se retornan al llamante, y cesa el procesamiento de la petición.

Si se efectúa la protección de los resultados, no se efectuará la fusión de los resultados.

- 6) Si se trata de una operación de modificación, continuar en el paso 7).

Si se trata de una operación de interrogación de una sola inserción, continuar en el paso 8).

Si se trata de una operación de interrogación de múltiples inserciones, continuar en el paso 9).



- 7) Cuando se sigue un procedimiento de modificación, puede ser preciso establecer, modificar o concluir vinculaciones operacionales, o bien puede ser preciso actualizar las sombras a consecuencia de la ejecución de la operación. El que esto se haga de forma síncrona o asíncrona con la ejecución de la operación original depende de las respectivas operaciones de modificación (y de las decisiones a nivel local). Se retorna al llamante un resultado o error local o DAP/DSP.
- 8) El resultado de una operación de interrogación de una sola inserción se retorna directamente al llamante como un resultado local o DAP/DSP.
- 9) Si la operación es una operación de interrogación de múltiples inserciones comprobar la **nameResolutionPhase** de la operación. Si no está **completed** llamar el procedimiento **List(I)** o **Search(I)**, y en otro caso llamar el procedimiento **List(II)** o **Search(II)**, respectivamente.
- 10) El resultado final de una llamada al procedimiento **List(II)** (resultado o error) y el resultado final de una llamada al procedimiento **List(I)** (en caso de que el resultado final sea un error) puede retornarse directamente al originador de la llamada (como resultado local o DAP/DSP).

Si el procedimiento llamado fue el procedimiento **List(I)**, el resultado podría contener referencias de continuación que tienen que ser desreferenciadas (en función de los controles de servicio y de las decisiones a nivel local). Esto puede determinar el envío de List operations (operaciones de listar) concatenadas a los DSA respectivos. Para fusionar los resultados, proseguir en el paso 5) con la llamada del procedimiento **Result Merging**.

- 11) Si la operación fue una operación Search (buscar), cualesquiera referencias de continuación se resuelven por el procedimiento **Search Continuation Reference** (si se requiere y está permitido). Esto puede hacer que se envíen peticiones Search (buscar) concatenadas a los respectivos DSA. Se llama el procedimiento **Result Merging** [paso 5)] para fusionar los resultados de búsqueda y quizás desreferenciar la referencia de continuación contenida, si existe.

### 16.3 Visión de conjunto de los procedimientos

Esta cláusula presenta una visión de conjunto de la funcionalidad básica de los procedimientos utilizados por el **despachador de operaciones** que se definen en las cláusulas 17 a 22.

#### 16.3.1 Procedimiento Request Validation (validación de petición)

Este procedimiento, descrito en la cláusula 17, se llama antes de efectuar una resolución de nombre local, para realizar una comprobación de bucles, de límites y de seguridad. Este procedimiento también proporciona valores por defecto para aquellos parámetros del **ChainingArgument** no proporcionados por el DAP, en el caso de que la petición procediese de un DUA. Además, este procedimiento detecta cualquier petición **abandon** y la notifica al **despachador de operaciones**.

#### 16.3.2 Procedimiento Abandon (abandonar)

Este procedimiento, descrito en 20.5, intenta hallar la operación que ha de abandonarse y la termina. Si existen subpeticiones pendientes, pueden enviarse tras ellas operaciones Chained Abandon. El procedimiento o retorna al llamante un **Null Result**, o una indicación de error (por ejemplo, **abandonError** con el problema **tooLate**).

#### 16.3.3 Procedimiento Find DSE (encontrar DSE)

Este procedimiento, descrito en 18.2 y 18.3 coteja los componentes del nombre del objeto pretendido con las DSE contenidos localmente para resolver el nombre de objeto pretendido. Sí se encuentra una DSE de alias, se desreferencia el alias (si se permite) y se reinicia el procedimiento para resolver el nuevo nombre.

Si no se encontró el objeto pretendido, se prosigue el procedimiento con el subprocedimiento **Target Not Found**. Si se encontró, se prosigue el procedimiento en el subprocedimiento **Target Found**.

NOTA – **Target Not Found** y **Target Found** son continuaciones del procedimiento **Find DSE**.

El procedimiento puede dar por resultado diversos errores, en cuyo caso el error de protocolo asociado se devuelve al solicitante y se termina el **despachador de operaciones**.

##### 16.3.3.1 Subprocedimiento Target Not Found (objeto de destino no encontrado)

Este procedimiento, descrito en 18.3.2, realiza una evaluación de las DSE inmediatas localizadas y crea un conjunto de **referencias de continuación** en **NRcontinuationList**, sobre la base del conjunto de referencias de conocimiento que se han detectado dentro del procedimiento **Find DSE**. Este conjunto de referencias se procesa ulteriormente dentro del procedimiento **Name Resolution Continuation Reference**.

El procedimiento puede dar lugar a diversos errores, en cuyo caso se devuelve al solicitante el error de protocolo asociado y se termina el despachador de operaciones.

### 16.3.3.2 Subprocedimiento Target Found (tarjet encontrado)

Este procedimiento, definido en 18.3.3, comprueba si la DSE encontrada es idónea para la operación solicitada, por ejemplo, en el caso en que se trate de información sombreada. Esto puede incluir la comprobación de la idoneidad de todo el subárbol de información sombreada por debajo del objeto de destino en el caso de una operación de objetos múltiples (por ejemplo, una búsqueda de subárbol).

Si la inserción localizada es idónea se invoca el procedimiento de evaluación de la operación apropiado. En otro caso se crea una **ContinuationReference** que apunta al suministrador (o maestro de la información) en **NRcontinuationList** y se invoca el procedimiento **Name Resolution Continuation Reference**.

### 16.3.4 Procedimiento de interrogación de una sola inserción

Este procedimiento, descrito en 19.2, se invoca para ejecutar efectivamente las operaciones que afectan a una sola inserción, es decir **Read (leer)** y **Compare (comparar)**. Una vez concluido, se retorna al DSA/DUA solicitante una respuesta (resultado o error) creada por el procedimiento.

### 16.3.5 Procedimientos de modificación

Estos procedimientos, descritos en 19.1, se ejecutan para procesar las operaciones de modificación, es decir, **AddEntry** (añadir inserción), **RemoveEntry** (suprimir inserción), **ModifyEntry** (modificar inserción) y **ModifyDN** (modificar DN). Para ello se ejecuta un subprocedimiento específico definido para cada una de estas operaciones. Durante (o después de) estos subprocedimientos se pueden enviar a otros DSA peticiones conformes a los protocolos DOP y DISP. Tras una conclusión exitosa, se retorna al DSA/DUA solicitante un resultado (creado por los subprocedimientos).

### 16.3.6 Procedimientos interrogación de múltiples inserciones

Estos procedimientos, descritos en 19.3, se ejecutan para procesar operaciones que afectan a varias inserciones que pueden estar o no ubicadas en el mismo DSA. Esto se realiza ejecutando subprocedimientos específicos definidos para cada una de las operaciones **Search** y **List** para realizar la descomposición de petición. Estos procedimientos crean un resultado local de la evaluación de la operación y opcionalmente un conjunto de referencias de continuación en **SRcontinuationList**. Si **SRcontinuationList** está vacío al final de este procedimiento, el resultado creado se retorna directamente al DSA/DUA solicitante. En otro caso, estas referencias de continuación son procesadas invocando el procedimiento **List** o **Search Continuation Reference**, según el tipo de operación.

### 16.3.7 Procedimiento Name resolution Continuation Reference (referencia de continuación de la resolución del nombre)

Este procedimiento, descrito en 20.4.1, procesa las referencias de continuación en la **NRcontinuationList** creada durante la fase de resolución del nombre. Estas referencias de continuación o se utilizan para emitir subpeticiones concatenadas o se retornan en un error de referimiento. En el caso de la concatenación, los resultados o errores retornados a partir de la petición concatenada se retornan para su procesamiento ulterior por el procedimiento **Result Merging** (fusión de resultados).

### 16.3.8 Procedimiento List and Search Continuation Reference (referencia de continuación de listar y buscar)

Estos procedimientos, descritos en 20.4.3 y 20.4.4, procesan las referencias de continuación en la **SRcontinuationList** creada por los procedimientos de interrogación de inserciones múltiples y los resuelven emitiendo subpeticiones concatenadas o creando referencias de continuación dentro de **partialOutcomeQualifier.unexplored**. Cuando se han recibido los resultados o errores de todas las subpeticiones pendientes, se devuelven para su procesamiento ulterior por el procedimiento de **fusión de resultados**.

### 16.3.9 Procedimiento Result Merging (fusión de resultados)

Este procedimiento, descrito en la cláusula 21, examina el resultado desde una petición concatenada o combina los resultados de operaciones locales con los resultados recibidos de subpeticiones concatenadas. Si una subpetición hubiese retornado un error, este procedimiento determina cómo ha de manejarse este error.

Si quedan referencias de continuación en el resultado, serán desreferenciadas (si lo permite la normativa local y lo exigen los controles de servicio) por el procedimiento **Name Resolution**, **List** o **Search Continuation Reference**, según los casos. Si no está firmado, se retiran los duplicados del resultado.

El resultado fusionado (con todos los resultados fusionados y las referencias de continuación no resueltas) se retorna al DUA/DSA solicitante.

Si se efectúa la protección de los resultados, no se efectuará la fusión de los resultados.

## 17 Procedimiento de validación de petición

### 17.1 Introducción

El procedimiento de **validación de petición** es el punto de entrada del despachador de operaciones para entradas (inputs) procedentes de DUA y de DSA. Este procedimiento prepara las entradas (inputs) para el procesamiento de la resolución de nombre. La función de este procedimiento consiste en detectar las operaciones de abandonar, realizar comprobaciones de seguridad, ajustar las entradas recibidas de los DUA para que puedan procesarse de la misma forma que las entradas recibidas de los DSA, comprobar la validez de la sintaxis y la semántica de los argumentos de la petición de validez, realizar la detección de bucles y otras diversas comprobaciones. El flujo de la **validación de la petición** se describe en la figura 8.

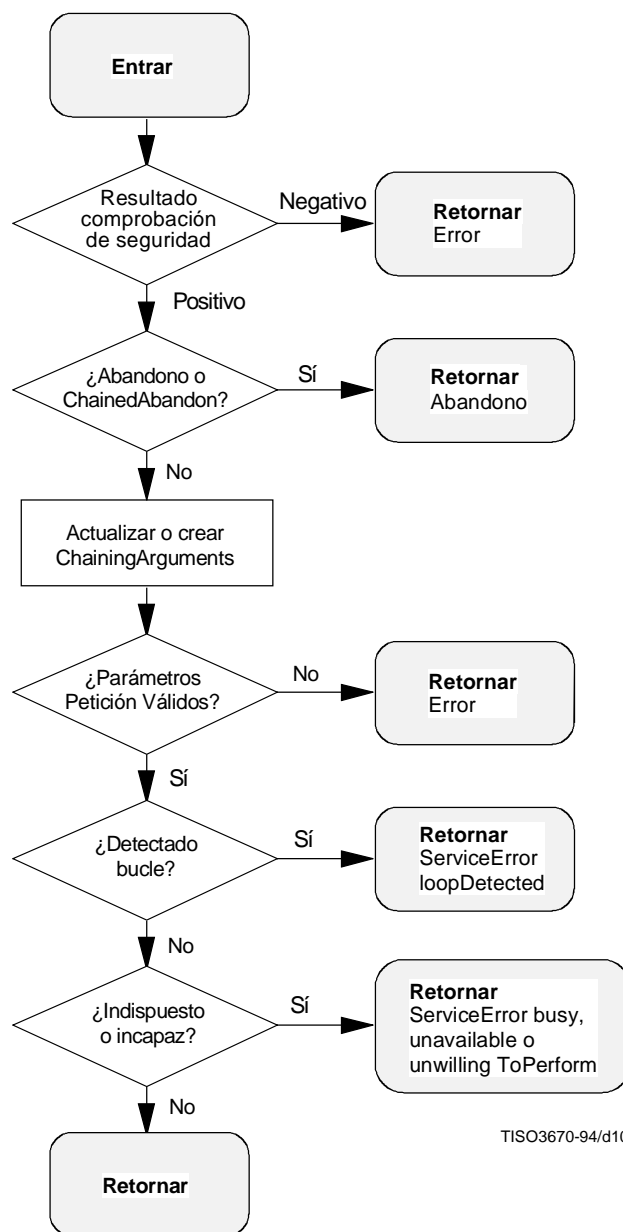


Figura 8 – Procedimiento Request Validation (validación de petición)

## 17.2 Parámetros del procedimiento

### 17.2.1 Argumentos

El argumento de entrada de **validación de petición** se compone de **ChainingArguments** (excepto en el caso de las operaciones **chainedAbandon**), si la petición se recibe de un DSA, y el argumento enviado por el originador de la petición.

### 17.2.2 Resultados

El resultado de salida (output) de **validación de petición** puede tener cinco formas diferentes.

- a) Si fracasa la comprobación de seguridad, se retorna al solicitante un error.
- b) Si la entrada es una operación **abandon** o **chainedAbandon** la salida es el argumento de la operación.
- c) Si los argumentos de la petición no son válidos se retorna un error al solicitante. En función de la política local, el DSA puede optar por retornar un **serviceError** o un **securityError**.
- d) Si se detecta un bucle, se retorna al solicitante un **serviceError** con el problema **loopDetected**.
- e) Si, motivado por problemas de recursos o consideraciones de política, el DSA no es capaz o no está dispuesto a realizar la operación, se retorna al solicitante un **serviceError** (con el problema **busy**, **unavailable**, o **unwillingToPerform**).
- f) En todos los demás casos, la entrada validada, transformada por adición de **ChainingArguments** si se recibe de un DUA o la actualización de **ChainingArguments.traceInformation** si se recibe de un DSA, es la salida del procedimiento y, por consiguiente, la entrada a la **resolución de nombre**.

## 17.3 Definición del procedimiento

Se realiza la comprobación de seguridad descrita en 17.3.2. Esto puede dar por resultado el retorno de un error y la terminación del despachador de operaciones.

Si la entrada es una operación **abandon** o **chainedAbandon**, sólo se realizan posteriormente los pasos establecidos en 17.3.1, y en otro caso se realizan los pasos indicados en 17.3.3 a 17.3.5. 17.3.5 describe el procedimiento de detección de bucles que puede dar por resultado el retorno de un error y la terminación del despachador de operaciones.

A continuación se realizan las comprobaciones de 17.3.6. Pueden tener por resultado en el retorno de un error y la terminación del despachador de operaciones.

Si las comprobaciones de 17.3.2 a 17.3.6 no culminan en la terminación del despachador de operaciones, se realizan los pasos de 17.3.7 y el procedimiento termina con la transferencia de su salida al procedimiento **Name Resolution** (**resolución de nombre**).

### 17.3.1 Procesamiento de un abandono

El argumento de un **abandon** o **chainedAbandon** se pasa al procedimiento **Abandon** (véase 20.5), para procesar la petición de abandono.

### 17.3.2 Comprobaciones de seguridad

Si el argumento de la operación está firmado, criptado o firmado y criptado, se ha de comprobar la signatura. Si esta fuese no válida o la descriptación fallara, o faltase en un caso en el que debía estar presente, puede retornarse un error al solicitante. De otro modo, un DSA puede realizar cualesquiera otras acciones definidas localmente.

### 17.3.3 Preparación del input

#### 17.3.3.1 Petición de DUA

Si la operación se recibe de un DUA, se crea un valor **ChainingArguments** de la manera siguiente.

- a) Se fija **ChainingArguments.originator** según se describe en 10.3.
- b) **ChainingArguments.operationProgress** se fija al valor de **CommonArguments.operationProgress**.

- c) **ChainingArguments.tracelInformation** se fija a una secuencia que contenga un único valor **Traceltem**. Este valor se construye del siguiente modo. **Traceltem.dsa** se fija al nombre del DSA que ejecuta la validación de petición. Deberá omitirse **Traceltem.targetObject**. **Traceltem.operationProgress** se fija en el valor entrante.
- d) Si el control de servicio de la operación especifica un límite de tiempo (el tiempo, en segundos disponible para la compleción de la operación), **ChainingArguments.timeLimit** se fija al tiempo UTC en el cual la operación deberá estar concluida para que cumpla el límite de tiempo especificado por el usuario.
- e) **ChainingArguments.AuthenticationLevel** y **ChainingArguments.UniqueIdentifier** se fijan según la política de seguridad local.
- f) Si se fija la opción **manageDSAIT** en **ServiceControls**, entonces:
  - el componente **nameResolutionPhase** de **operationProgress** se pondrá para ser **completado**;
  - el componente **nexRDNTToBeResolved** de **operationProgress** se omitirá;
  - **referenceType** tomará el valor **self**;
  - **entryOnly** tomará el valor **FALSE**;
  - **nameResolveOnMaster** tomará el valor **FALSE**; y
  - se fijará la opción **chainingProhibited** en **ServiceControls**;
  - se omiten los restantes elementos facultativos de **ChainingArguments**, y se adoptan valores por defecto cuando se especifican.
- g) Si no se pone la opción **manageDSAIT** en **ServiceControls**, los restantes elementos facultativos de **ChainingArguments** se omiten, y se adoptan valores por defecto cuando se especifican.
- h) **ChainingArguments.SecurityParameters.ProtectionRequest** se utiliza para indicar el nivel de protección (firma, criptación, o firma y criptación) que se ha de conferir a los resultados.

### 17.3.3.2 Petición de DSA

Si se recibe la operación desde un DSA, se actualiza la **ChainingArguments.tracelInformation** agregando un valor al final de la secuencia **Traceltem**. Este valor se construye del siguiente modo.

- a) **Traceltem.dsa** se fija al nombre del DSA que está ejecutando la validación de la petición.
- b) **Traceltem.targetObject** se fija al valor de **ChainingArguments.targetObject** a menos que el **object** (o el **baseObject** en el caso de la búsqueda) del argumento de petición sea idéntico a **ChainingArguments.targetObject**, en cuyo caso se omitirá **Traceltem.targetObject**.
- c) **Traceltem.operationProgress** se fija al valor de **ChainingArguments.operationProgress**.

### 17.3.4 Aserción de validez

Se comprobará la operación para verificar la validez de la sintaxis y de la semántica de sus argumentos de acuerdo con las reglas de las cláusulas que definen cada operación (por ejemplo debe comprobarse que el **nextRDNTToBeResolved** no proporciona un número que supera el número de RDN del **targetObject**.) Si se detecta que la petición contiene argumentos no válidos, se termina la operación y se retorna un error al usuario, según la invalidez detectada.

### 17.3.5 Detección de bucle

Si cualesquiera dos valores **Traceltem** de **ChainingArguments.tracelInformation** (preparada como se indicó en 17.3.3) son idénticos, el procesamiento de la operación ha retornado a un estado anterior, es decir se ha detectado un bucle. En este caso se retornará un **serviceError** (con el problema **loopDetected**) al solicitante y terminará el despachador de operaciones.

### 17.3.6 Incapaz o indispuerto a actuar

**Request Validation** (validación de petición) puede evaluar los recursos disponibles y determinar que no puede realizarse la operación. También puede determinar, sobre la base de consideraciones de política, que no debe realizarse la operación. En estos casos puede retornarse un **serviceError** (con el problema **busy**, **unavailable**, o **unwillingToPerform**) al solicitante y con ello termina el despachador de operaciones.

### 17.3.7 Procesamiento de la salida

En la fase final de la **validación de petición**, la salida validada, transformada por adición de **ChainingArguments** si se recibe de un DUA o la actualización de **ChainingArguments.traceInformation** si se recibe de un DSA, se retorna y se utiliza como entrada al procedimiento de **resolución de nombre**.

## 18 Resolución de nombre

### 18.1 Introducción

Esta cláusula describe el procedimiento **Name Resolution (resolución de nombre)**, sus argumentos, resultados y posibles condiciones de error. Según se muestra en la figura 16 (despachador de operaciones), el procedimiento de **resolución de nombre** comprende dos procedimientos:

- Procedimiento **Find DSE (encontrar DSE)**.
- Procedimiento **Name Resolution Continuation Reference (referencia de continuación de resolución de nombre)**.

El procedimiento de **Find DSE** se describe en tres diagramas de flujo, a saber **Find DSE**, **Target Found**, y **Target Not Found**. El procedimiento **Find DSE** coteja el nombre de la inserción de destino con las DSE almacenadas localmente, componente por componente. Si se encuentra localmente la inserción de destino, **Find DSE** prosigue con el subprocedimiento **Target Found**, que entonces llama al procedimiento **Check Suitability (comprobar idoneidad)** para comprobar la idoneidad de la DSE hallada para la evaluación. Si no se encuentra localmente la inserción de destino, **Find DSE** prosigue con el subprocedimiento **Target Not Found** y prepara la(s) referencia(s) de continuación que se añadirá a **NRcontinuationList** para que el procedimiento **Name Resolution Continuation Reference** la despache.

NOTA 1 – La **resolución de nombre** cotejará el nombre con los múltiples valores distinguidos diferenciados por el contexto, según se describe en 9.4 de la Rec. UIT-T X.501 | ISO/CEI 9594-2, al determinar una correspondencia.

NOTA 2 – La **resolución de nombre** puede fallar si un DSA superior anterior a 1997 contiene una referencia subordinada a una inserción mantenida en una edición DSA posterior, y el RDN para esa inserción incluye contextos. La **resolución de nombre** fallará frente a una copia sombra de una inserción cuando se utilice un nombre alternativo como nombre pretendido y la inserción sombra se mantenga en una primera o segunda edición del DSA.

### 18.2 Parámetros del procedimiento Find DSE

#### 18.2.1 Argumentos

El procedimiento utiliza los siguientes argumentos:

- a) **ChainingArguments.aliasDereferenced;**
- b) **ChainingArguments.aliasedRDNs;**
- c) **ChainingArguments.excludeShadows;**
- d) **ChainingArguments.nameResolveOnMaster;**
- e) **ChainingArguments.operationProgress (nameResolutionPhase, nextRDNTToBeResolved);**
- f) **ChainingArguments.referenceType;**
- g) **ChainingArguments.targetObject;**
- h) el tipo de operación;
- i) el argumento de operación;

NOTA – Donde no existan valores explícitos deberán usarse valores por defecto, como se especifica en 10.3.

#### 18.2.2 Resultados

**Find DSE** tiene dos casos de resultado final exitoso (indicados por **entry suitable** o por **entry unsuitable** (inserción idónea/no idónea):

El primero de ellos retorna (desde el subprocedimiento **Target Not Found**) referencia(s) de continuación en **NRcontinuationList** que a continuación se pasa(n) al procedimiento **Name Resolution Continuation Reference** para continuar con la fase de resolución del nombre.

## ISO/CEI 9594-4 : 1998 (S)

El segundo caso de resultado exitoso retorna (desde el subprocedimiento **Target Found**) un(a) (referencia a un) DSE, que se pasa a uno de los procedimientos de evaluación.

### 18.2.3 Errores

Pueden retornarse los siguientes errores:

- a) **serviceError: unableToProceed, invalidReference, unavailableCriticalExtension;**
- b) **nameError: noSuchObject, aliasDereferencingProblem, contextProblem.**

### 18.2.4 Variables globales

El procedimiento utiliza las siguientes variables globales:

- **NRcontinuationList**; esta es una lista para almacenar la(s) referencia(s) de continuación necesaria(s) para continuar la resolución de nombre en el procedimiento **Name Resolution Continuation Reference**.

### 18.2.5 Variables locales y compartidas

El procedimiento utiliza las siguientes variables locales:

- a) **i** Índice utilizado para identificar el componente del nombre de destino sobre el que se trabaja.
- b) **m** La longitud del nombre de objeto de destino que va a utilizarse en una resolución de nombre. Para las operaciones que efectúan una resolución de nombre con respecto a la inserción progenitora (parent entry), esto es, AddEntry, **m** se fija a (número de RDN en el objeto de destino) – 1. Para todas las demás operaciones, **m** se fija al número de RDN en el objeto de destino.
- c) **lastEntryFound** Índice, así, DSE(lastEntryFound) es el último DSE cotejado que es del tipo **entry**.
- d) **lastCP** Índice, así, DSE(lastCP) es el último prefijo de contexto sombreado encontrado.
- e) **candidateRefs** Un conjunto de referencias de continuación.

También se utiliza la variable compartida **admPoints** (definida en el despachador de operaciones). Por razones de comodidad, el componente **i** del nombre de objeto de destino se designa por **N(i)**.

## 18.3 Procedimientos

NOTA – En el diagrama de flujo hay algunos textos que sólo son pertinentes para operaciones específicas. Esto no se indica en los diagramas de flujo, pero se describe en el texto que los acompaña.

### 18.3.1 Procedimiento Find DSE

Véase la figura 9.

Este procedimiento intenta resolver localmente el nombre del objeto de destino.

- 1) Inicializar las variables locales **lastEntryFound** y **lastCP** a **0**; **admPoints** y **candidateRefs** a un conjunto vacío, e inicializar **i** a **0**.
- 2) Comparar **i** con **m**. Si no son iguales, continuar en el paso 5).
- 3) Si son iguales, comprobar si la **nameResolutionPhase** está **completed**. Si no está **completed**, continuar en el subprocedimiento **Target Not Found**.

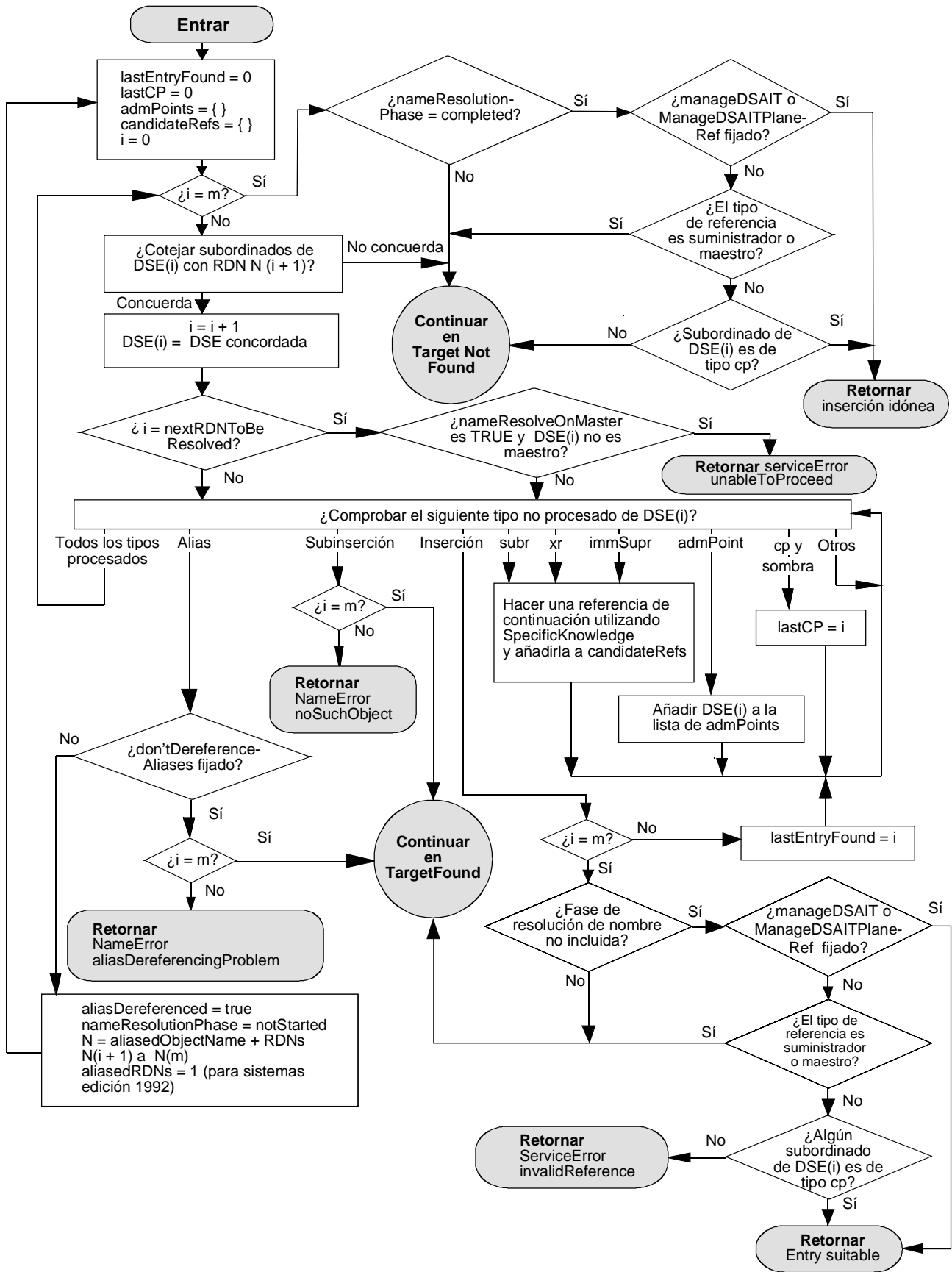
Si ya se ha terminado la **fase de resolución de nombre** y se pone la extensión crítica **manageDSAIT**, retornar con **inserción adecuada**.

- 4) Si **nameResolutionPhase** está **completed**, comprobar si cualquier inmediatamente subordinada de DSE(i) es un prefijo de contexto (del tipo **cp**).

Si una (o más) DSE(s) inmediatamente subordinadas es del tipo **cp**, retornar con **entry suitable**.

NOTA 1 – Este caso es para las subpeticiones **List (II)** y **Search (II)**.

Si ninguna de las inmediatamente subordinadas de DSE(i) es de tipo **cp**, continuar en el subprocedimiento **Target Not Found**.



TISO8930-99/d11

Figura 9 – Procedimiento Find DSE (encontrar DSE)



- 5) Intentar encontrar una concordancia para el componente ( $i + 1$ ) del nombre de objeto de destino con el nombre de un subordinado de la última DSE concordada. En el caso de  $i = 0$ , intentar cotejar uno de las DSE inmediatamente subordinados al DSE raíz. Si no puede encontrarse concordancia, continuar en el subprocedimiento **Target Not Found**. Si se encontró una sola concordancia, incrementar  $i$  y almacenar la DSE concordada como el elemento  $i$  del vector de las DSE encontradas.

NOTA 2 – La concordancia de nombre incluye el tratamiento de múltiples valores distinguidos diferenciados por el contexto, cuando se conocen, según se describe en 9.4 de la Rec. UIT-T X.501 | ISO/CEI 9594-2.

Si se encontró más de una concordancia, retornar el **NameError** con problema **contextProblem**.

NOTA 3 – Este puede ser el caso, por ejemplo, cuando un **AttributeTypeAndDistinguishedValue** en un nombre pretendido contiene múltiples valores de atributo distinguidos diferenciados por los contextos y varios de esos valores coinciden con valores en diferentes nombres de destino.

- 6) Si  $i$  es igual a **nextRDNTToBeResolved**, comprobar si se satisfacen las dos condiciones siguientes:

- el **ChainingArgument.nameResolveOnMaster** es **TRUE**,
- DSE( $i$ ) no es una inserción maestra.

Si se satisfacen ambas condiciones, retornar **error de servicio** con problema **unableToProceed**.

NOTA 4 – Esto indica que se ha utilizado **nameResolveOnMaster** para evitar múltiples caminos al mismo objeto de destino.

- 7) Comprobar todos los bits de tipo DSE de DSE( $i$ ). Para el bit de cada tipo, puede requerirse algún procesamiento. A continuación se indica la acción a continuar para cada tipo encontrado:

- Si los bits **cp** y **shadow** están fijados, recordar el índice  $i$  del **lastCP**.
- Si el bit **admPoint** está fijado, comprobar el atributo operacional **administrativeRole**. Si éste es el comienzo de un área administrativa autónoma, vaciar la lista **admPoints**. Si éste es el comienzo de una o más áreas administrativas específicas, comprobar la lista **admPoints** y suprimir los puntos existentes que ya no ofrezcan interés (es decir, aquellos puntos cuyos roles hayan sido reemplazados por el nuevo punto administrativo). Almacenar DSE( $i$ ) en la lista.
- Si está fijado uno de los bits **subr**, **xr** o **immSupr**, generar una referencia de continuación utilizando el atributo **specificKnowledge** con **operationProgress.nameResolutionPhase** fijado a **procedimiento**, **nextRDNTToBeResolved** fijado a  $i$ , **targetObject** construido a partir de los componentes resueltos utilizando RDN primarios (pueden incluirse valores distinguidos alternativos en los RDN) concatenados con el resto de los componentes no resueltos, y **accessPoints** y **referenceType** fijados como correspondan. Añadir la referencia de continuación a la lista de referencias de continuación en **candidateRefs**.
- Si está fijado el bit **entry**, comprobar si  $i$  es igual a  $m$  (y por lo tanto el nombre de objeto de destino completamente está concordado). Si  $i$  no es igual a  $m$ , recordar la inserción encontrada fijando **lastEntryFound** a  $i$  y continuar procesando los bits de tipo de DSE( $i$ ). Si  $i$  y  $m$  son iguales, continuar en el paso 8).
- Si está fijado el bit **subentry**, comprobar si  $i$  es igual a  $m$  (y por lo tanto el nombre de objeto de destino totalmente está concordado). Si son iguales, continuar en el procedimiento **Target Found**; si no son iguales, retornar un **nameError** con el problema **noSuchObjet**.
- Si está fijado el bit de **alias**, comprobar si **dontDereferenceAliases** está fijado.

Si no está fijado **dontDereferenceAliases**, puede desreferenciarse el alias. Por ello, fijar **chainingArguments.aliasDereferenced** a **TRUE**, **nameResolutionPhase** a **notStarted**, y el nombre del objeto de destino al **aliasedEntryName** que se proporciona en la inserción de alias concatenado con los componentes restantes no concordados del nombre de objeto de destino anterior [es decir concatenar con los componentes del ( $i + 1$ ) al  $m$  del nombre de objeto de destino anterior]. Los DSA edición 1988 no fijan **aliasedRDNs** a 1 (mientras que los DSA de la edición de 1988 fijan los **aliasedRDNs** al número de RDN en **aliasedEntryName**). Comenzar de nuevo la **resolución de nombre** continuando en el paso 1).

Si está fijado **dontDereferenceAliases**, el alias no puede desreferenciarse. Comprobar si el nombre del objeto de destino se ha procesado por completo comprobando si  $i$  y  $m$  son iguales. Si son iguales (y por tanto el nombre está totalmente concordado) continuar en el subprocedimiento **Target Found**. Si no son iguales (y por tanto el nombre no está totalmente concordado) retornar **nameError** con el problema **aliasDereferencingProblem**.

- Para todos los demás tipos posibles de DSE no es necesaria acción alguna. Marcar internamente ese tipo de DSE como procesado y continuar procesando los bits de tipo de DSE no procesados de la DSE(i).
  - Si están procesados todos los bits tipo de la DSE(i), continuar en el paso 2).
- 8) Comprobar si **nameResolutionPhase** está **completed**. En caso contrario, continuar en el subprocedimiento **Target Found**.
- 9) Si **nameResolutionPhase** está **completed**, comprobar si el **referenceType** utilizado es **supplier** o **master**; en este caso, continuar en el subprocedimiento **Target Found**.
- NOTA 5 – Esto se aplica a las subpeticiones de concatenar a suministrador.
- Si la **fase de resolución de nombre** ya está **completa** y se ha fijado la extensión crítica **manageDSAIT**, retornar con **inserción adecuada**.
- 10) En caso contrario, comprobar si cualquiera de las DSE inmediatamente subordinadas a DSE(i) es un prefijo de contexto (y por ello de tipo **cp**). Si alguna(s) lo es (son), retornar **entry suitable**. Si ninguna de las inserciones inmediatamente subordinadas es del tipo prefijo de contexto, retornar un **serviceError** con el problema **invalidReference**.
- NOTA 6 – Este caso es para las peticiones **List (II)** y **Search (II)**.

### 18.3.2 Subprocedimiento Target Not Found (inserción de destino no encontrado)

Véase la figura 10.

Se llama a este subprocedimiento cuando en el DSA local no se encuentra el nombre de objeto de destino. Determina el mejor tipo de referencia de conocimiento que habrá de utilizarse para continuar la resolución de nombre, a menos que se detecte un error, en cuyo caso se retorna ese error.

- 1) Cuando se prosigue desde el procedimiento **Find DSE**, se distinguen tres posibles casos de fase de resolución de nombre.
  - Si **nameResolutionPhase** está **notStarted**, continuar en el paso 2).
  - Si **nameResolutionPhase** está **proceeding**, continuar en el paso 8).
  - Si **nameResolutionPhase** está **completed**, continuar en el paso 12).
- 2) Si se encontró alguna inserción (**lastEntryFound** no igual a 0), fijar **nameResolutionPhase** a **proceeding** y continuar en el paso 9).
- 3) Si no se encontró ninguna inserción (**lastEntryFound=0**), comprobar si el DSA es un DSA de primer nivel.
  - Si es un DSA de primer nivel, la DSE raíz no contiene una referencia superior y por ello no es de tipo **supr**. En este caso, continuar en el paso 4).
  - Si no es un DSA de primer nivel, la DSE raíz contiene una referencia superior y por ello es de tipo **supr**. En este caso, generar una referencia de continuación utilizando el conocimiento superior encontrado en la DSE raíz. Fijar:
    - **targetObject** al nombre del objeto de destino construido a partir de los componentes resueltos utilizando los RDN primarios (en los RDN se pueden incluir valores distinguidos alternativos) concatenados con los restantes componentes no resueltos;
    - **operationProgress.nameResolutionPhase** a **notStarted**;
    - **referenceType** a **superior**; y
    - **accessPoints** como corresponda.
 Añadir la referencia de continuación a la lista de referencias de continuación en **candidateRefs**. Continuar en el paso 6).
- 4) Comprobar si la operación fue dirigida a la inserción raíz (**m = 0?**). Si lo fue, continuar en el paso 5). En caso contrario, generar una referencia de continuación utilizando cualquier conocimiento NSSR encontrado en la DSE raíz. Fijar:
  - **targetObject** al nombre del objeto de destino construido a partir de los componentes resueltos utilizando los RDN primarios (en los RDN se pueden incluir valores distinguidos alternativos) concatenados con los restantes componentes no resueltos;
  - **operationProgress.nameResolutionPhase** a **proceeding**;

- **operationProgress.nextRDNTToBeResolved** a 1;
- **referenceType** a **nonSpecificSubordinate**; y
- **accessPoints** como corresponda.

Añadir la referencia de continuación a la lista de referencias de continuación en **candidateRefs**. Continuar en el paso 6).

- 5) En un DSA de primer nivel, sólo pueden realizarse operaciones de listar o buscar con la inserción raíz como objeto de base. Por ello, si la operación no fue de List (listar) o Search (buscar), retornar **nameError** con el problema **noSuchObject**. Si fue una operación de listar o buscar, fijar **nameResolutionPhase** a **completed** y retornar con **entry suitable**.
- 6) Comprobar si hay alguna referencia de continuación en el **candidateRefs**. Si **candidateRefs** está vacío y **partialNameResolution** es **FALSE**, retornar **nameError** con el problema **noSuchObject**. Si **candidateRefs** está vacío y **partialNameResolution** es **TRUE**, entonces en el resultado fijar **partialName** a **TRUE**, **nameResolutionPhase** a **completed**, y retornar con **entry suitable**. De otro modo, proseguir en el paso 7).
- 7) Utilizar una función de selección local para elegir una referencia de continuación de la lista de referencias de continuación en **candidateRefs**, añadirla a la lista de referencias de continuación de **NRcontinuationList**, y retornar con **entry unsuitable**.
- 8) Si el DSA no fue capaz de proseguir con (resolución de nombre) (en cuyo caso **lastEntryFound** es menor que **nextRDNTToBeResolved**), continuar en el paso 11). En otro caso continuar en el paso siguiente.
- 9) Si DSE(i) es una DSE de sombra con conocimiento subordinado incompleto (**subordinateCompletenessFlag** es **FALSE**), generar una referencia de continuación desde el atributo **supplierKnowledge** encontrado en **DSE(lastCP)**. Fijar:
  - **targetObject** al nombre del objeto de destino construido a partir de los componentes resueltos utilizando RDN primarios (en los RDN se pueden incluir valores distinguidos alternativos) concatenados con los restantes componentes no resueltos;
  - **operationProgress.nameResolutionPhase** a **proceeding**;
  - **operationProgress.nextRDNTToBeResolved** a **lastEntryFound**;
  - **referenceType** a **supplier**; y
  - **accessPoints** como corresponda.

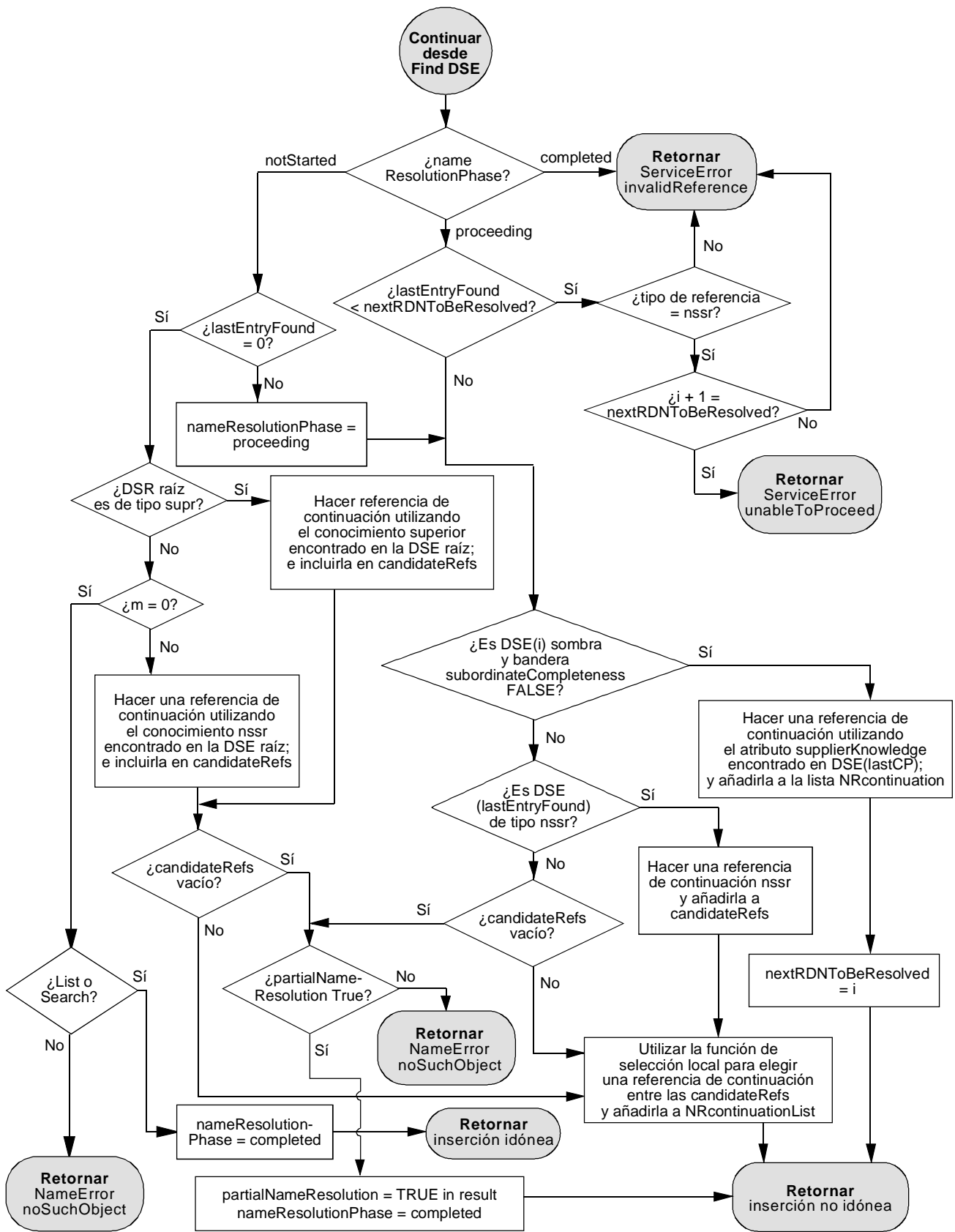
Añadir la referencia de continuación a la lista de referencias de continuación de **NRcontinuationList**, y retornar con **entry unsuitable**.

- 10) Si la última inserción encontrada contiene un NSSR [**DSE(lastEntryFound)** es de tipo **nssr**], generar una referencia de continuación a partir del conocimiento NSSR encontrado en **DSE(lastEntryFound)**. Fijar:
  - **targetObject** al nombre del objeto de destino construido a partir de los componentes resueltos utilizando los RDN primarios (en los RDN se pueden incluir valores distinguidos alternativos) concatenados con los restantes componentes no resueltos;
  - **operationProgress.nameResolutionPhase** a **proceeding**;
  - **operationProgress.nextRDNTToBeResolved** a **lastEntryFound+1**;
  - **referenceType** a **nonSpecificSubordinate**; y
  - **accessPoints** como corresponda.

Añadir la referencia de continuación a la lista de referencias de continuación en **candidateRefs**. Continuar al paso 7).

Si **DSE(lastEntryFound)** no es de tipo **nssr**, continuar en el paso 6).

- 11) Si **chainingArguments.referenceType** es de tipo **nssr**, continuar en el paso 13), y en otro caso en el paso 12).
- 12) Retornar **serviceError** con el problema **invalidReference**.
- 13) Si **i + 1** es igual a **nextRDNTToBeResolved**, la petición fue encaminada hacia este lugar debido a una NSSR y el DSA es incapaz de proseguir con la resolución de nombre; en este caso, retornar **serviceError** con el problema **unableToProceed**; en otro caso, continuar en el paso 12).



TISO8940-99/d12

Figura 10 – Subprocedimiento Target Not Found (inserción de destino no encontrado)

**18.3.3 Subprocedimiento Target Found (asiento de destino encontrado)**

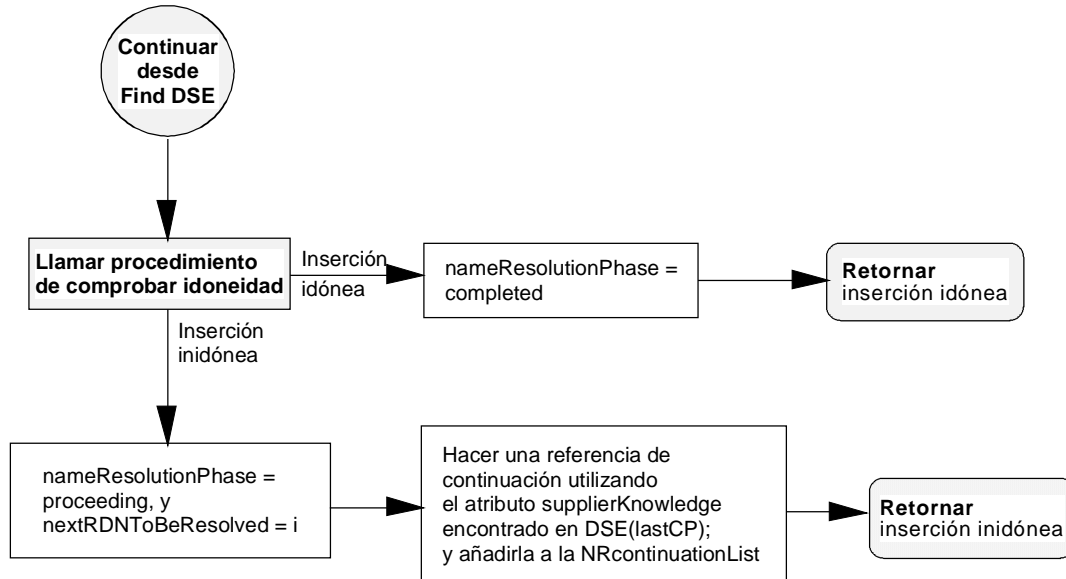
Se entra en este subprocedimiento cuando el nombre del objeto de destino concuerda localmente con un inserción de DSE. Este subprocedimiento comprueba si la inserción encontrada es idónea para procesar localmente la petición. Se muestra en la figura 11.

- 1) Llamar el procedimiento de comprobación de idoneidad.
- 2) Si la inserción es idónea (**entry suitable**), fijar **nameResolutionPhase** a **completed** y retornar **entry suitable**.
- 3) Si la inserción es inidónea (**entry unsuitable**), generar una referencia de continuación utilizando el atributo **supplierKnowledge** encontrado en **DSE(lastCP)**. Fijar:
  - **targetObject** al nombre del objeto de destino construido a partir de los componentes resueltos utilizando los RDN primarios (en los RDN se pueden incluir valores distinguidos alternativos) concatenados con los restantes componentes no resueltos;
  - **operationProgress.nameResolutionPhase** a **proceeding**;
  - **operationProgress.nextRDNTToBeResolved** a **m**;
  - **referenceType** a **supplier**; y
  - **accesPoints** como corresponda.

Añadir la referencia de continuación a la lista de referencias de continuación de **NRcontinuationList**. Retornar **entry unsuitable**.

NOTA – Sin embargo, si el **ServiceControl localScope** es **TRUE**, el DSA podría decidir, por razones de política local, considerar esta inserción como idónea y proseguir como en el paso 2).

- 4) Si una extensión crítica no es soportada (**unsupported critical extensión**), retornar **serviceError** con el problema **unavailableCriticalExtension**.



TISO3700-94/d13

**Figura 11 – Subprocedimiento Target Found (inserción de destino encontrado)**

**18.3.4 Procedimiento Check Suitability (comprobar idoneidad)**

Se llama este procedimiento para decidir si una DSE encontrada es idónea para realizar la operación solicitada (véase la figura 12). Tiene en cuenta los **ChainingArguments**, los **ServiceControls**, los argumentos proporcionados por el usuario, el tipo de operación y las características de la DSE (sombra, conocimiento subordinado, atributos presentes, etc.).

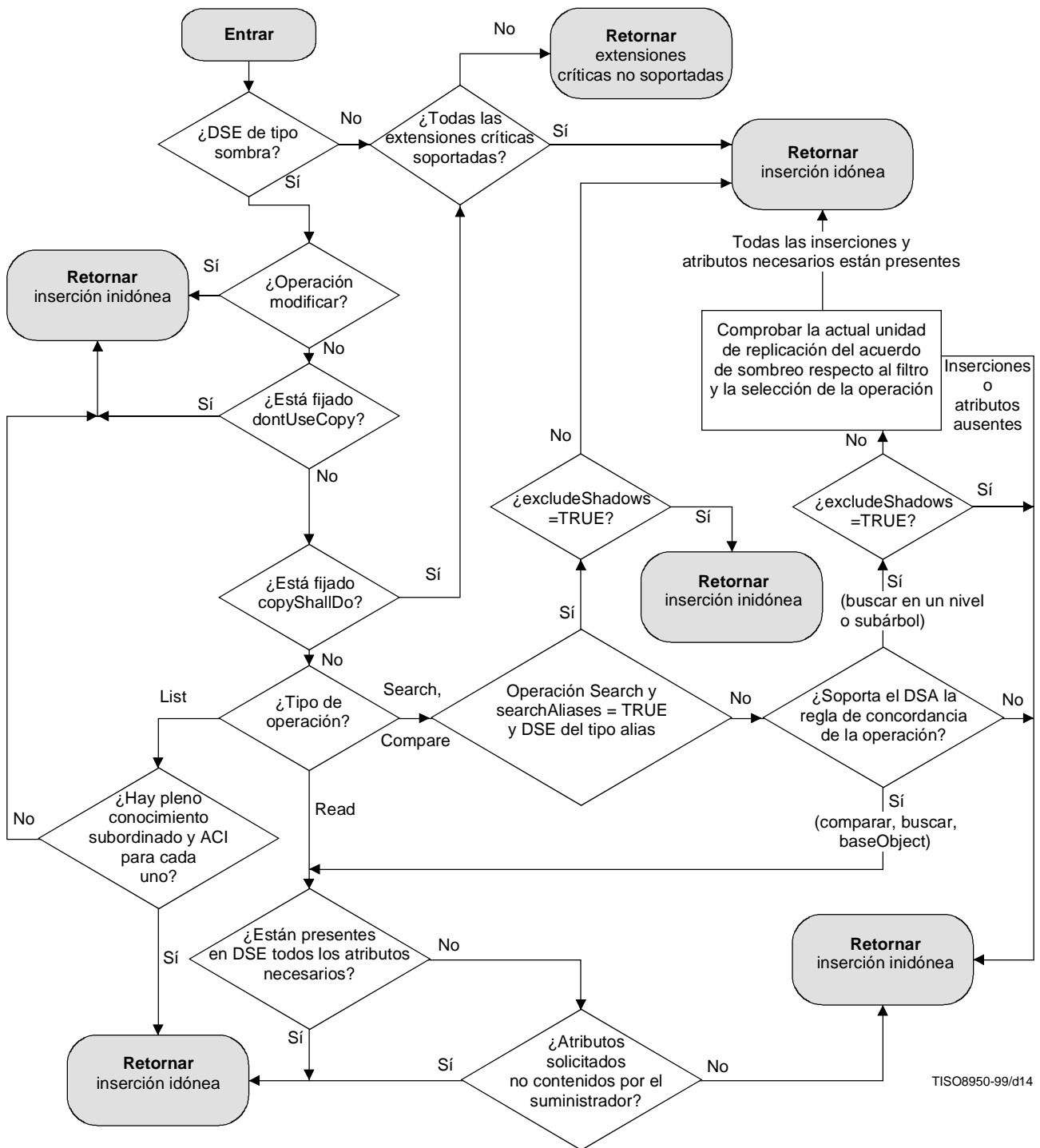


Figura 12 – Procedimiento Check Suitability (comprobar idoneidad)

### 18.3.4.1 Parámetros del procedimiento

Los argumentos de entrada de este procedimiento son:

- una referencia a una DSE;
- el tipo de operación para el que va a comprobarse la idoneidad de la DSE;
- los **ChainingArguments**; y
- el argumento de operación.

La salida es o bien **inserción idónea** o **inserción inidónea**, o **extensión crítica no soportada**.

- 1) Si la DSE no es de tipo **shadow**, comprobar si todas las **criticalExtensions** son soportadas. Si lo son, retornar una **inserción idónea**; si no, retornar extensión crítica no soportada.
- 2) La DSE es de tipo **sombra**. Retornar inserción inidónea, si se da cualquiera de los dos casos siguientes:
  - La operación solicitada es de tipo modificación.
  - Está fijado el control de servicio **dontUseCopy**.
 En otros casos, continuar en el paso siguiente.
- 3) Si está fijado el control de servicio **copyShallDo**, comprobar si todas las **criticalExtensions** son soportadas. Si lo son, retornar entonces **inserción idónea**; si no, retornar **extensión crítica no soportada**.
- 4) Si no está fijado el control de servicio **copyShallDo**, comprobar si todas las **criticalExtensions** son soportadas. Si lo son, continuar en el paso 5); si no, retornar **inserción inidónea**.
- 5) Distinguir entre tipos de operación:
  - Si se trata de la operación List (listar), continuar en el paso 6).
  - Si se trata de la operación Read (leer), continuar en el paso 7).
  - Si se trata de las operaciones Search (buscar) o Compare (comparar), continuar en el paso 8).
- 6) Si la inserción tiene un conocimiento subordinado pleno, puede realizarse la operación List. En este caso, retornar **inserción idónea**, y en otro caso retornar **inserción inidónea**.
- 7) Si en la DSE están presentes todos los atributos solicitados, retornar **inserción idónea**. Si faltan algunos atributos, determinar por medios locales si la copia sombra contiene todos los atributos que contenía la inserción maestra (por ejemplo, por referencia a un convenio de sombreado). Si los contiene, la inserción es idónea (retornar **inserción idónea**). De no ser así, el suministrador puede tener los atributos solicitados que no están presentes en la sombra; en este caso, la petición ha de concatenarse (retornar **inserción inidónea**).
- 8) Si la operación es **search** con **searchAliases** puesto a **TRUE** y el DSE es del tipo **alias**, entonces si **ChainingArguments.excludeShadows** es **FALSE** retornar **entry suitable**, si es **TRUE** retornar **entry unsuitable**.
- 9) Si el DSA soporta la regla de concordancia para la comparación o búsqueda que se solicita y la operación es una operación **compare** o **search** con **subset** de **baseObject**, continuar en el paso 7). Si el DSA soporta la regla de concordancia y la operación es **search** con el subset (subconjunto) **oneLevel** o **subtree**, continuar en el paso 9). En otro caso retornar **inserción inidónea**.
- 10) Si **ChainingArguments.excludeShadows** es **TRUE**, retornar **inserción inidónea**. En otro caso, comprobar la comprensión local de la especificación de información sombreada con el filtro y la selección de la operación. Si están presentes todos los inserciones y atributos necesarios, retornar **inserción idónea**. Si falta alguna inserción o atributo, retornar **inserción inidónea**.

## 19 Evaluación de la operación

Esta cláusula define al procedimiento que deberá seguir un DSA si se ha hallado localmente la inserción de destino de una operación (durante la resolución de nombre). Según el tipo de operación, se invoca uno de los siguientes procedimientos:

- Para una operación **addEntry**, **chainedAddEntry**, **removeEntry**, **chainedRemoveEntry**, **modifyEntry**, **chainedModifyEntry**, **modifyDN** o **chainedModifyDN** se continuarán los procedimientos de 19.1.
- Para una operación **read**, **chainedRead**, **compare** o **chainedCompare** se continuarán los procedimientos de 19.2.
- Para una operación **search**, **chainedSearch**, **list** y **chainedList** se continuarán los procedimientos de 19.3.

## 19.1 Procedimiento de modificación

Según el tipo de operación de modificación se seguirán los correspondientes procedimientos definidos en 19.1.1 a 19.1.4.

### 19.1.1 Operación Add Entry (añadir inserción)

- 1) El DSA comprobará que el iniciador tiene suficientes derechos de acceso, por ejemplo según se define en 11.1.5 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. En caso contrario, se retorna el correspondiente error.
- 2) El DSA se asegurará de que no existe ya una inserción con el nombre del asiento a añadir, y en caso contrario retornará un **updateError** con el problema **entryAlreadyExists**. Si la DSE superior es de tipo adicional **nssr**, el DSA continuará el procedimiento definido en 19.1.5 (Operaciones de modificación y NSSR) para asegurar que el nombre de la nueva inserción es inequívoca. Si el nombre de la inserción que se ha de añadir incluye múltiples valores distinguidos diferenciados por el contexto para algún atributo en el RDN final, el DSA asegurará que ninguno de los posibles RDN alternativos que se puedan construir genere un nombre (independientemente del contexto) para una inserción que ya existe.
- 3) Si **targetSystem** está presente, y el **AccessPoint** no es el del DSA actual, continuar en el paso 4). Si **targetSystem** no está presente, o si está presente y el **AccessPoint** es el del DSA actual, continuar en el paso 5).
- 4) Si la inserción es una subinserción, el DSA retornará **updateError** con el problema **affectsMultipleDSAs**. Si la inserción no es una subinserción, el DSA tiene la opción local de establecer o no, a su voluntad, una HOB con el DSA especificado. Si opta por no establecerla, el DSA retornará **serviceError** con el problema **unwillingToPerform**. Si opta por establecerla, el DSA establecerá una vinculación operacional jerárquica (HOB) con el DSA subordinado especificado. Si el DOP está soportado, deberá seguirse el procedimiento descrito en 24.3.1.1. En caso contrario se utilizarán medios locales para establecer la HOB. Si el DSA subordinado no desea establecer la vinculación operacional, se retorna un **serviceError** con el problema **unwillingToPerform** para la operación **addEntry**. Si tiene éxito el establecimiento de la HOB, continuar en el paso 7).

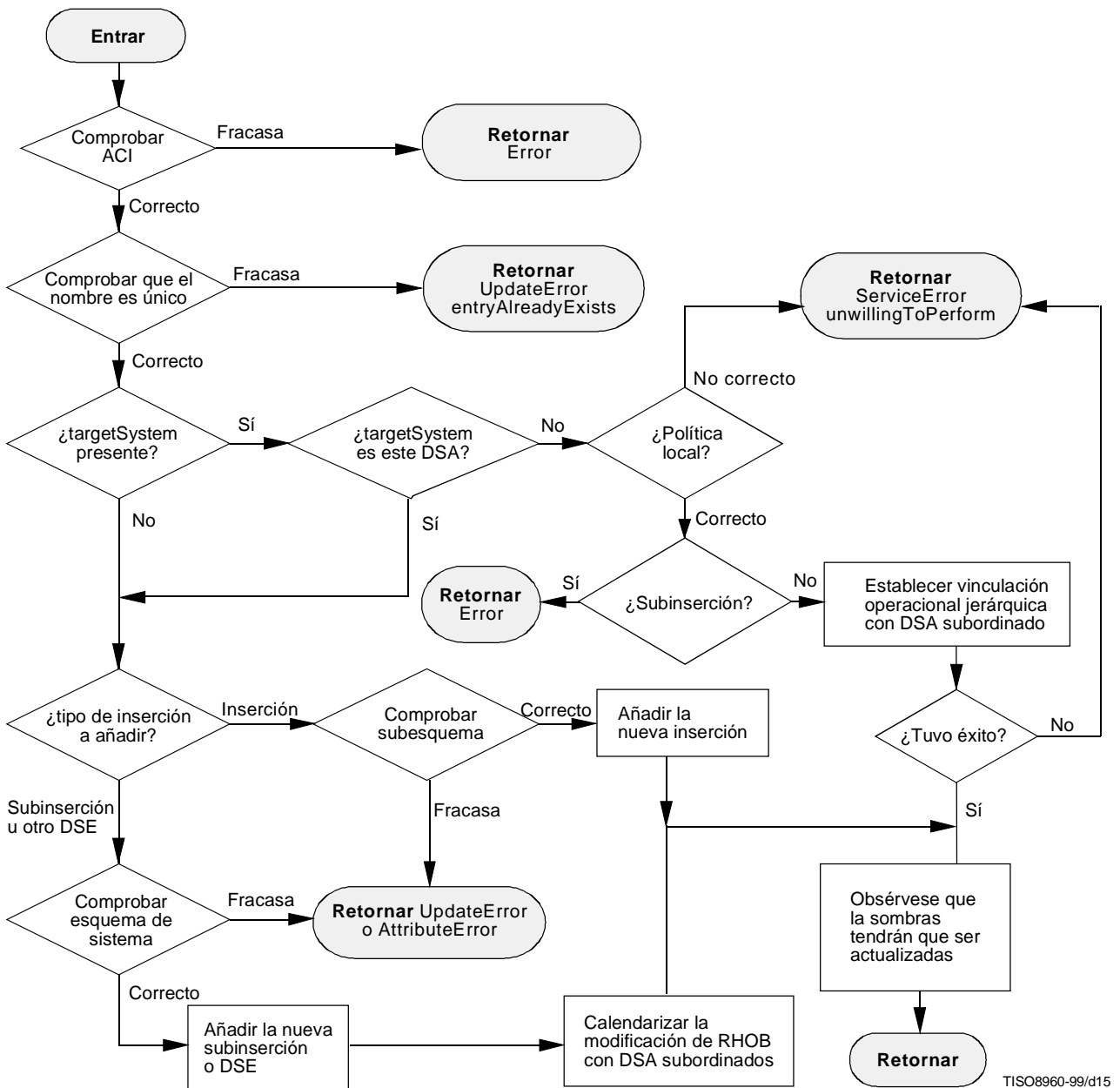
NOTA 1 – Este paso del procedimiento no se aplica a la creación de áreas administrativas autónomas en un DSA subordinado.

- 5) El DSA asegurará que la nueva inserción está en consonancia con el subesquema, o que la nueva subinserción o DSE de otros tipos está en conformidad con el esquema del sistema (por ejemplo, que el DSE superior inmediato de una subinserción es del tipo **admPoint**). En caso contrario, retornará un **updateError** o **attributeError** adecuado, o bien añadirá el nuevo DSE. Si se trata de una inserción, continuar en el paso 7). Si se trata de una subinserción, continuar en el paso 6). De otro modo, se ejecutan procedimientos de gestión de conocimiento adecuados para los otros tipos de DSE. Véase la sección 6, Administración de conocimiento.
- 6) El DSA enviará, en un momento apropiado, una vinculación operacional de modificar a todos los DSA subordinados relevantes con los cuales tenga vinculaciones operacionales jerárquicas o jerárquicas no específicas. Son vinculaciones relevantes aquellas que están asociadas con contextos de denominación que están subordinados a la DSE superior. Los contextos de denominación cuyos prefijos de contexto corresponden a puntos administrativos autónomos no son relevantes. Si el DOP está soportado, deberán seguirse los procedimientos de 24.3.2.1 y 25.3.2. Si el DOP no está soportado, deberán utilizarse medios locales para modificar las RHOB.

NOTA 2 – El momento (o tiempo) apropiado lo especifica el administrador del DSA, y pudiera estar comprendido entre el instante que sigue (o incluso que precede) inmediatamente a aquel en que se retorna el resultado de la operación a una estrategia periódica (por ejemplo, a una hora convenida). Este momento (o tiempo) puede variar según el motivo de la modificación, por ejemplo las actualizaciones de ACI son de efecto inmediato y los cambios del esquema se efectúan periódicamente.

- 7) Si la inserción o subinserción añadida está dentro de la **UnitOfReplication** de uno o más acuerdos de sombreado, los consumidores de sombra deberán ser actualizados utilizando los procedimientos del servicio de sombra de información de directorio especificado en la Rec. UIT-T X.525 | ISO/CEI 9594-9.





TISO8960-99/d15

Figura 13 – Procedimiento Add Entry (añadir inserción)

19.1.2 Operación Remove Entry (suprimir inserción)

- 1) El DSA comprobará que el iniciador tiene suficientes derechos de acceso, por ejemplo los definidos en 11.2.5 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. En caso contrario, se retorna un error apropiado.
- 2) El DSA se asegurará de que la inserción a suprimir es un inserción hoja. En otro caso el DSA retornará un **updateError** con el problema **notAllowedOnNonLeaf**.
- 3) Se comprueba el tipo DSE de la inserción que se desea suprimir. Si es **subentry**, continuar en el paso 5). Si es **cp**, continuar en el paso 6). Si es **entry** o **alias**, continuar en el paso 4). De otro modo, se ejecutan procedimientos de gestión de conocimiento adecuados para los otros tipos de DSE. Véase la sección 6.
- 4) Suprimir la inserción, o inserción de alias, y continuar en el paso 7).
- 5) Suprimir la inserción. En un momento apropiado, modificar las vinculaciones operacionales de todos los DSA subordinados relevantes con los cuales el actual DSA tiene vinculaciones operacionales jerárquicas o jerárquicas no específicas. Son vinculaciones relevantes aquellas que están asociadas con contextos de denominación subordinados a la DSE superior.

Los contextos de denominación cuyos prefijos de contexto corresponden a puntos administrativos autónomos no son relevantes. Si el DOP está soportado, deberán seguirse los procedimientos indicados en 24.3.2.1 y 25.3.2. Si no lo está deberán utilizarse medios locales. Continuar en el paso 7).

- 6) Suprimir el contexto de denominación. Si el DSA tiene una vinculación operacional jerárquica para este contexto de denominación, terminará la vinculación operacional jerárquica con su DSA inmediatamente superior. Si el DSA tiene una vinculación operacional jerárquica no específica para este contexto de denominación, y este es el último contexto de denominación de la vinculación operacional jerárquica no específica, terminará la vinculación operacional no específica con su DSA inmediatamente superior. Si el DOP está soportado, deberán seguirse los procedimientos indicados en 24.3.3.2 y 25.3.3.2. Si no está deberán utilizarse medios locales para terminar la RHOB.
- 7) Si el contexto de denominación, inserción, inserción de alias o subinserción suprimidos estaba dentro de la **UnitOfReplication** de uno o más de los acuerdos de sombreado, los consumidores de sombra serán actualizados mediante los procedimientos del servicio de sombra de información de directorio especificados en la Rec. UIT-T X.525 | ISO/CEI 9594-9.

Si la referencia subordinada o referencia subordinada no específica suprimida en el DSA inmediatamente superior (cuya RHOB fue terminada) estaba dentro de la **UnitOfReplication** de uno o más acuerdos de sombreado, los consumidores de sombra serán actualizados mediante los procedimientos del servicio de sombra de información del directorio especificados en la Rec. UIT-T X.525 | ISO/CEI 9594-9.

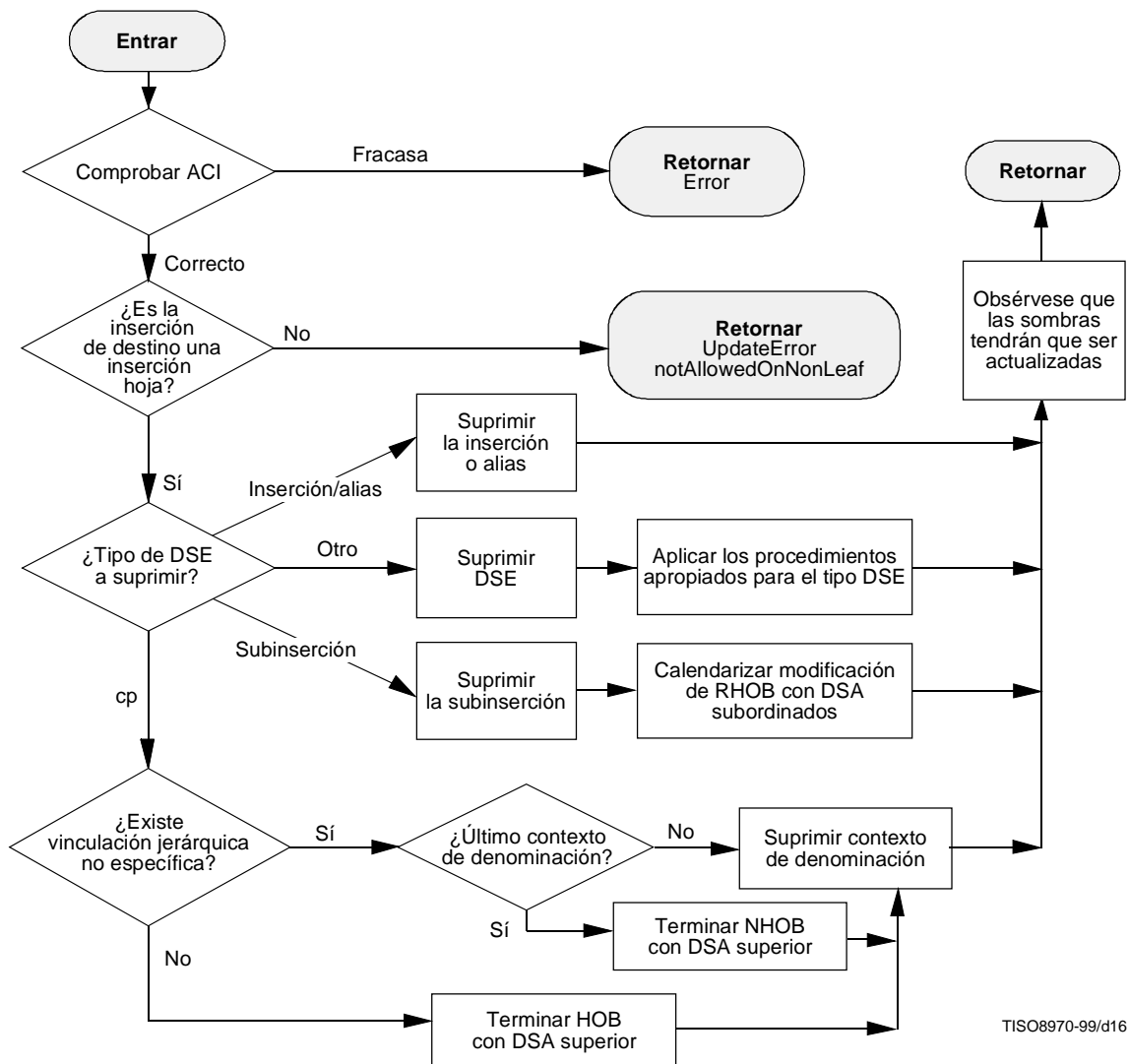


Figura 14 – Procedimiento Remove entry (suprimir inserción)

19.1.3 Operación modificar asiento

- 1) El DSA comprobará que el iniciador tiene derechos de acceso, por ejemplo como los definidos en 11.3.5 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. En caso contrario se retornará un error apropiado.
- 2) Las modificaciones de la inserción o alias estarán en conformidad con el subesquema. La modificación de una DSE de otros tipos, incluida la subinserción, estará en conformidad con el esquema del sistema. De otro modo, el DSA retornará un **updateError** o **attributeError** apropiado. Una vez efectuadas las modificaciones, si el DSE de destino es del tipo **subentry**, continuar en el paso 3); si la DSE de destino es del tipo **entry** o **alias**, continuar en el paso 4); de otro modo, se ejecutan procedimientos de gestión de conocimiento apropiados para los otros tipos DSE. Véase la sección 6.
- 3) El DSA, en un momento apropiado, modificará las vinculaciones operacionales con todos los DSA subordinados relevantes con los que tiene vinculaciones operacionales jerárquicas o jerárquicas no específicas. Son vinculaciones relevantes aquellas que están asociadas con contextos de denominación que están subordinados al punto administrativo debajo del cual está situada la subinserción modificada. Los contextos de denominación cuyos prefijos corresponden a puntos administrativos autónomos no son relevantes. Si el DOP está soportado, deberá seguirse el procedimiento indicado en 24.3.2.1 y 25.3.2. Si no lo está deberán emplearse medios locales.
- 4) Si la inserción modificada, inserción de alias o subinserción estaba dentro de la **UnitOfReplication** de uno o más acuerdos de sombreado, los consumidores de sombra deberán actualizarse mediante los procedimientos del servicio de sombra de información del directorio especificado en la Rec. UIT-T X.525 | ISO/CEI 9594-9.

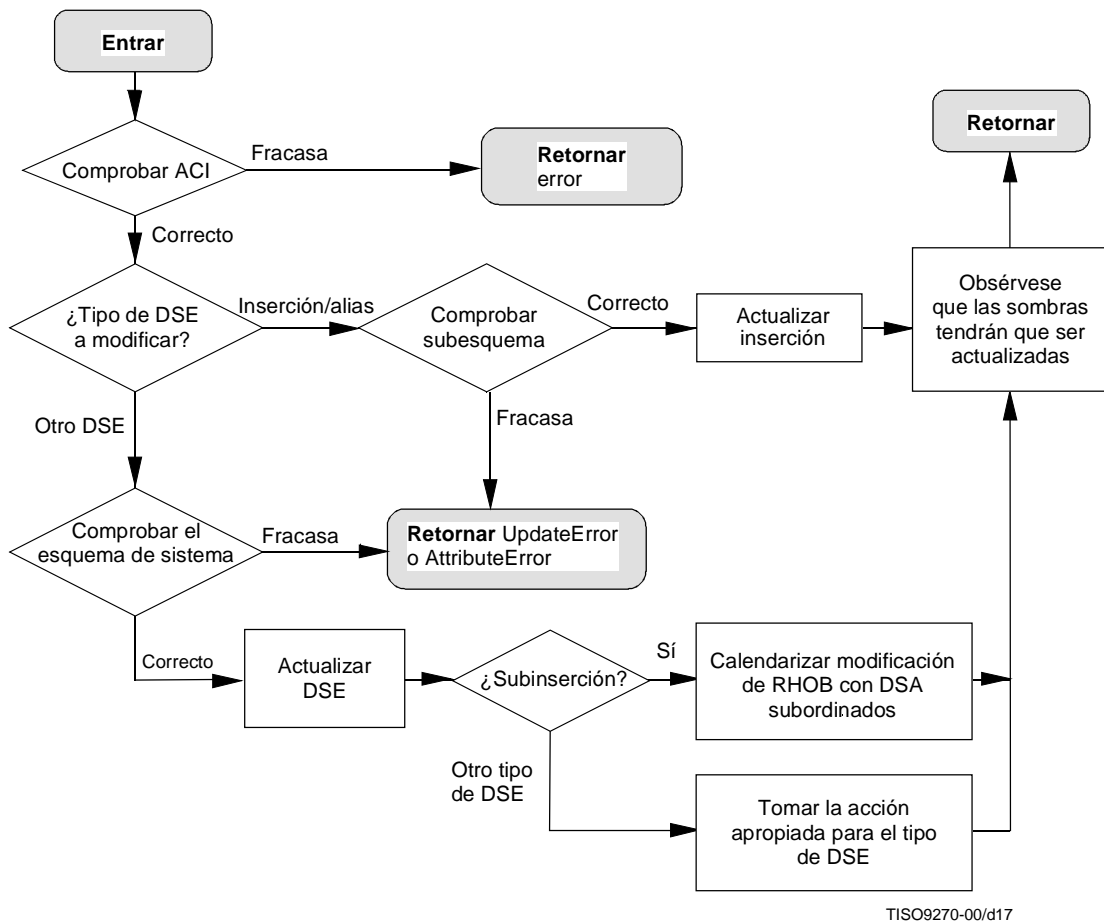


Figura 15 – Procedimiento Modify Entry (modificar inserción)

19.1.4 Operación Modify DN (modificar DN)

- 1) El DSA comprobará que el iniciador tiene suficientes derechos de accesos, por ejemplo, los definidos en 11.4.5 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. En caso contrario, se retorna un error apropiado.
- 2) Si la operación consiste en trasladar una inserción a un nuevo superior en el mismo DSA, continuar en el paso 3). Si consiste en cambiar el nombre distinguido relativo de una inserción, continuar en el paso 4).

- 3) La operación deberá ejecutarse de acuerdo con la definición que se da en 11.4.1 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. Si el antiguo superior, el nuevo superior, la inserción o cualquiera de sus subordinados están en otro DSA, o si el nuevo superior tiene NSSR, deberá rechazarse la operación retornando un **updateError** con el problema **affectsMultipleDSAs**. El DSA verificará que no existe ya otra inserción con el nuevo nombre. Si no lo es retornará un **updateError** con problema **entryAlreadyExist**. El DSA verificará que el nuevo nombre de la inserción es conforme al subesquema. Si no lo es, retornará un **attributeError** o **updateError**. De no ser así, trasladar las inserciones dentro del DSA y continuar en el paso 9).
- 4) El texto siguiente es aplicable a la modificación del nombre distinguido relativo de una inserción, la cual puede o no ser una inserción hoja, y puede o no tener uno o más subordinados en uno o más DSA. Se comprueba el tipo DSE de la inserción que va a ser red denominada. Si es **subentry**, continuar en el paso 7). Si es **cp**, continuar en el paso 6). Si es **entry** o **alias**, continuar en el paso 5).
- 5) El DSA verificará que no existe ya otra inserción con el nuevo nombre. Si existe, retornará un **updateError** con el problema **entryAlreadyExists**. Si la DSE superior de la inserción que va a ser red denominada es de tipo **nssr** adicional, el DSA deberá seguir el procedimiento definido en 19.1.5 (Operaciones de modificación y referencias subordinadas no específicas) para garantizar que el nuevo nombre de la inserción sea inequívoco. Si el nuevo nombre incluye múltiples valores distinguidos diferenciados por el contexto para algún atributo en un RDN, el DSA asegurará que ninguno de los posibles RDN que se puedan construir genere un nombre (independientemente del contexto) para una inserción que ya existe. El DSA verificará que el nuevo nombre de la inserción es conforme al subesquema. Si no lo es, retornará un **attributeError** o **updateError**. Redenominar la inserción, o la inserción de alias. Si se trata de una inserción no hoja que tiene subordinados en otros DSA, continuar en el paso 8); de no ser así, continuar en el paso 9).
- 6) El DSA verificará que el nuevo nombre del contexto de denominación es conforme al subesquema. Si no lo es, retornará un **attributeError** o **updateError**.

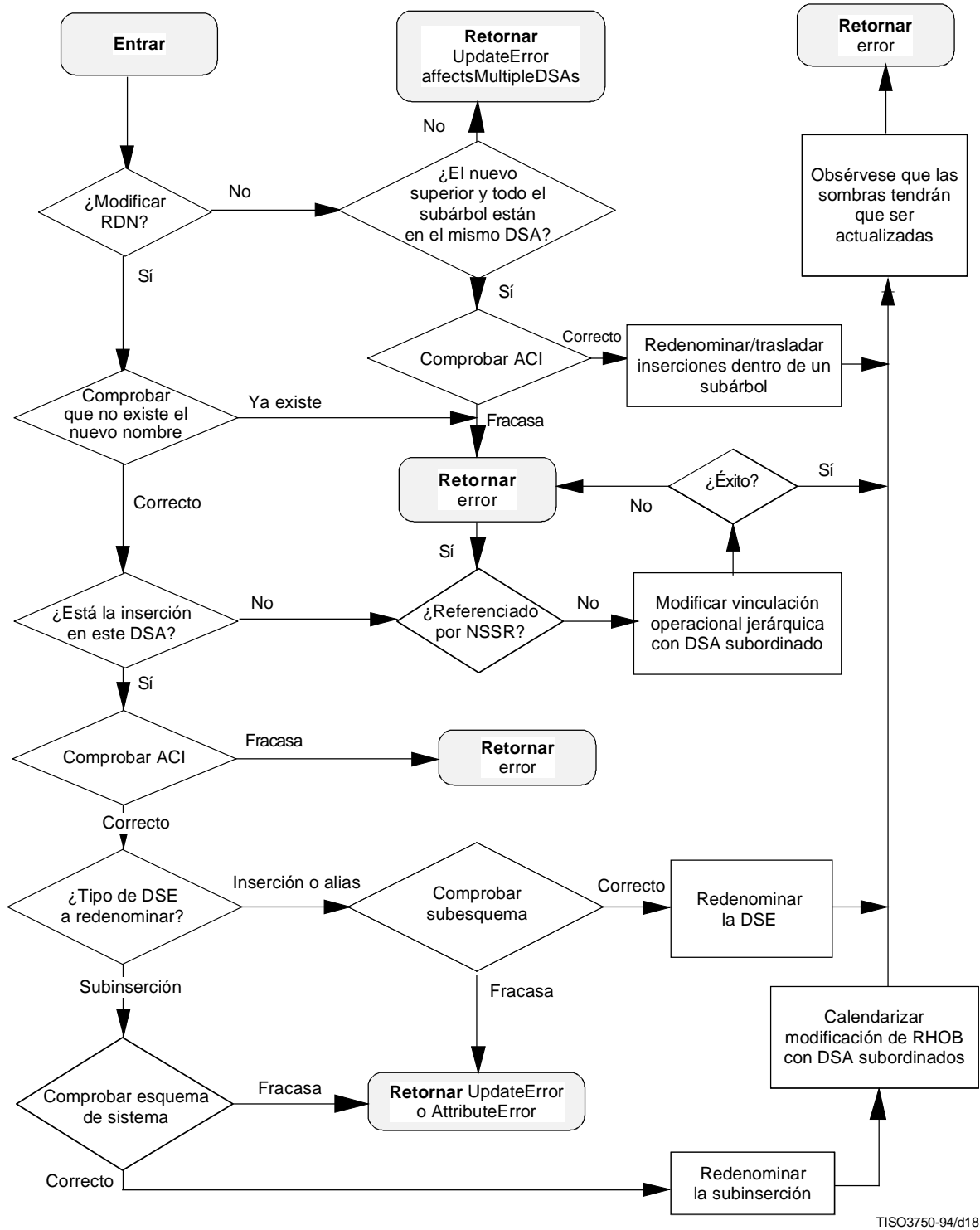
Si el DSA tiene una HOB con el DSA superior, el DSA subordinado tratará de modificar la HOB antes de responder a la operación de modificar DN. El DSA superior verificará que no existe ya otra inserción con el nuevo nombre, antes de aceptar la modificación. Si el DOP está soportado, deberá seguirse el procedimiento indicado en 24.3.2.2. Si el DOP no está soportado, la manera de modificar la HOB y de comprobar el nuevo nombre para determinar su unicidad es un asunto local. Si la HOB ha sido modificada con éxito y el contexto de denominación tiene contextos de denominación subordinados en otros DSA, continuar en el paso 8); en otro caso, continuar en el paso 9). Si no puede modificarse la HOB, retornar **updateError** con el problema **affectsMultipleDSAs**.

Si el DSA tiene una NHOB para este contexto de denominación con el DSA superior, la manera de detectar las inserciones duplicadas queda fuera del ámbito de esta Especificación de directorio. Redenominar la inserción. Si el contexto de denominación tiene contextos de denominación subordinados en otros DSA, continuar en el paso 8); si no, continuar en el paso 9).

- 7) El DSA se verificará de que el nuevo nombre de la subinserción es conforme al esquema de sistema, y si no lo es retornará un **attributeError** o **updateError** apropiado. El DSA asegurará que no exista ya otra subinserción con el nuevo nombre. Si existe retornará un **updateError** con el problema **entryAlreadyExists**.
- 8) El DSA, modificará, en un momento apropiado, las vinculaciones operacionales de todos los DSA subordinados relevantes con los cuales tiene vinculaciones operacionales jerárquicas o jerárquicas no específicas. Son vinculaciones relevantes aquellas que están asociadas con todos los contextos de denominación que están subordinados a la inserción que está siendo red denominada, o los contextos de denominación relevantes que están subordinados al punto administrativo cuya subinserción fue red denominada. Los contextos de denominación cuyos prefijos de contexto corresponden a puntos administrativos autónomos no son relevantes. Si el DOP está soportado, deberán seguirse los procedimientos descritos en 24.3.2.1 y 25.3.2. Si no deberán emplearse medios locales para actualizar las RHOB.
- 9) Si el contexto de denominación, inserción o cualquiera de sus subordinados, inserción de alias o subinserción red denominados está dentro de la **UnitOfReplication** de uno o más acuerdos de sombreado contenidos en el DSA, los consumidores de sombra deberán actualizarse mediante los procedimientos del servicio de sombra de información del directorio especificado en la Rec. UIT-T X.525 | ISO/CEI 9594-9.

Si la referencia subordinada red denominada en el DSA inmediatamente superior [cuya HOB fue modificada en el anterior paso 6)], está dentro de la **UnitOfReplication** de uno o más de sus acuerdos de sombreado, los consumidores de sombra deberán actualizarse mediante los procedimientos del servicio de sombra de información del directorio especificado en la Rec. UIT-T X.525 | ISO/CEI 9594-9.

Si unos componentes de una RHOB con un DSA subordinado [modificados como se ha indicado en el anterior paso 8)] están dentro de la **UnitOfReplication** de uno o más de los acuerdos de sombreado contenidos en el DSA subordinado, los consumidores de sombra deberán actualizarse mediante los procedimientos del servicio de sombra de información del directorio especificado en la Rec. UIT-T X.525 | ISO/CEI 9594-9.



TISO3750-94/d18

Figura 16 – Procedimiento Modify DN (modificar DN)

### 19.1.5 Operaciones de modificación y referencias subordinadas no específicas

Si un DSA tiene NSSR y no conoce el conjunto completo de nombres de los subordinados de una inserción a los cuales o bien,

- a) se ha dirigido una operación **addEntry**; o bien
- b) se ha dirigido una operación **modifyDN**;

el DSA ejecutará el siguiente conjunto de procedimientos antes de realizar la operación.

- 1) Si el control de servicio **chainingProhibited** está fijado en la operación **addEntry** o **modifyDN**, retornar **updateError** con el problema **affectsMultipleDSAs**.
- 2) Si el DSA no desea o no puede multiconcatenar peticiones salientes, retornar **serviceError** con el problema **unwillingToPerform** o **unavailable**, respectivamente.
- 3) El DSA concatenará una operación **chainedReadEntry** a cada DSA maestro en el conjunto de **accessPointInformation** del NSSR. (El DSA sólo utilizará el DSA maestro de cada **MasterAndShadowAccessPoints** debido a la inconsistencia transitoria producida por el sombreado.) Los parámetros del **ReadArgument** se fijarán del siguiente modo:

**object** o bien al nombre de la inserción a añadir (en el caso de **addEntry**) o al nombrepropuesto de una inserción existente (en el caso de **modifyDN**);

**selection** al atributo de clase de objeto.

Los parámetros de **CommonArguments** se fijarán del siguiente modo:

- **ServiceControls.options** fija **dontDereferenceAliases** a 1;
- **OperationProgress.nameResolutionPhase** se fija a **completed**.

Los parámetros de **ChainingArguments** se fijarán del siguiente modo:

- **originator** se fija al nombre del originador;
- **targetObject** se omite;
- **OperationProgress.nameResolutionPhase** se fija a **proceeding** y **nextRDNTobeResolved** a (número de RDN en el nombre de objeto) – 1;
- **traceInformation** se fija a una secuencia vacía;
- **referenceType** se fija a **nonSpecificSubordinate**;
- **timeLimit**, según sea adecuado conforme a la petición de entrada.

Otros parámetros, por ejemplo **SecurityParameters**, pueden fijarse como corresponda, por ejemplo, por razones de política local.

- 4) El DSA espera el conjunto completo de respuestas. Si una parte de la respuesta es un **ReadResult**, se retornará un error igual que en el paso 6) que sigue.
- 5) Si todas las respuestas son **serviceError** con el problema **unableToProceed**, puede proceder la evaluación de la operación.
- 6) Si se retorna un **ReadResult**, se retornará un **updateError** con el problema **entryAlreadyExists** para la operación original.
- 7) Si se retorna cualquier otro error a la petición **readEntry**, se retornará un **serviceError** con el problema **unwillingToPerform**.

El DSA que recibe la petición **chainedRead** dará una respuesta que dependerá de la presencia o no de la inserción y de su política de control de acceso.

## 19.2 Procedimiento de interrogación de una sola inserción

Las operaciones **read**, **chainedRead**, **compare** y **chainedCompare** entran en el grupo de procedimientos de interrogación de una sola inserción. Estos procedimientos comprenden sólo los tres pasos siguientes:

- 1) Comprobar el control de acceso según se describe en la cláusula 9 de la Rec. UIT-T X.511 | ISO/CEI 9594-3. Si la operación no está autorizada, retornar el correspondiente error de seguridad.
- 2) Realizar la operación en la DSE encontrado según se describe en la cláusula 9 de la Rec. UIT-T X.511 | ISO/CEI 9594-3.
- 3) Preparar la respuesta, y retornar.

### 19.3 Procedimiento de interrogación de múltiples inserciones

Según el tipo de operación de interrogación (**list** o **search**), se seguirán los correspondientes procedimientos definidos en 19.3.1 y 19.3.2 siguientes.

#### 19.3.1 Procedimiento List (listar)

Esta subcláusula especifica el procedimiento de evaluación específico a las operaciones **list** y **chainedList**.

El procedimiento **List (I)** se seguirá cuando el componente **operationProgress.nameResolutionPhase** de la petición de listar está fijado a **notStarted** o **proceeding** y cuando el DSA, tras realizar la resolución de nombre, encuentra que tiene el objeto base. El procedimiento **List (II)** se seguirá cuando el componente **nameResolutionPhase** de la petición de listar se ha fijado a **completed**.

##### 19.3.1.1 Parámetros del procedimiento

###### 19.3.1.1.1 Argumentos

Los argumentos utilizados por este procedimiento son:

- el **ListArgument**;
- la DSE de destino **e**;
- **operationProgress** del **chainingArgument**.

###### 19.3.1.1.2 Resultados

Si este procedimiento se ejecuta con éxito, retorna:

- un conjunto de subordinados de **e** en **listInfo.subordinates**;
- **limitProblem** indicado en **partialOutcomeQualifier**;
- un conjunto de referencias de continuación en **SRcontinuationList**.

###### 19.3.1.1.3 Errores

El procedimiento puede dar lugar a que se retorne uno de los siguientes errores al DUA/DSA solicitante:

- un **accessControlError** o **nameProblem**;
- cualquier error definido para el procedimiento **find DSE**, cuando se ha desreferenciado un alias.

##### 19.3.1.2 Definición del procedimiento

Los subprocedimientos definidos en 19.3.2.2.1 y 19.3.2.2.2 se invocarán según las siguientes reglas.

###### 19.3.1.2.1 Procedimiento List (I) [listar (I)]

El procedimiento **List (I)** se compone de los siguientes pasos, descritos en la figura 17.

- 1) Si el control de servicio **subentry** está fijado, continuar en el paso 5), de no ser así, continuar en el paso 2).
- 2) Si DSE **e** es de tipo **nssr**, añadir una continuation Reference a **SRcontinuationList** con los siguientes componentes:
  - **targetObject** al nombre distinguido primario del DSE **e** (en los RDN pueden incluirse valores distinguidos alternativos);
  - **aliasedRDNs** ausente;
  - **operationProgress** con **nameResolutionPhase** fijada a **completed** y **nextRDNtoBeResolved** ausente;
  - **rdnsResolved** ausente;
  - **referenceType** fijado a **nonSpecificSubordinate**;
  - **accessPoints** fijado a un conjunto de **accessPointInformation** derivada cada una de un valor de atributo **nonSpecificKnowledge** de la DSE **e**.
- 3) Para cada DSE **e'** inmediatamente subordinada a DSE **e**, ejecutar los siguientes pasos:
  - a) Comprobar la ACI en **e'** si está disponible. Si la ACI desautoriza el listado de RDN de **e'**, saltar esta DSE. Si la ACI no está disponible (por ejemplo en el caso de referencias subordinadas y glue), es cuestión de política local si se prosigue o no.

- b) Comprobar todos los tipos DSE de e'.
- i) Si e' es del tipo **subr**, se dan dos casos: En el primer caso, la ACI y la clase de objeto de la inserción subordinada están disponibles localmente, y entonces, sobre la base de la política local y del permiso para la ACI, añadir el RDN de e' a **listInfo.subordinates**, con **aliasEntry** fijado a **TRUE** si e' es de tipo **sa** y **fromEntry** fijado a **FALSE**. El segundo caso se da cuando la ACI de la inserción no está disponible en e', y entonces se añade una continuation Reference a **SRcontinuationList** con los siguientes componentes:
    - **targetObject** al nombre distinguido primario del DSE e (en los RDN pueden incluirse valores distinguidos alternativos);
    - **aliasedRDNs** ausente;
    - **operationProgress** con **nameResolutionPhase** fijado a **completed** y **nextRDNtoBeResolved** ausente;
    - **rdnsResolved** ausente;
    - **referenceType** fijado a **subordinate**;
    - **accessPoints** fijado al valor contenido en el atributo **specificKnowledge** de DSE e'.
  - ii) Si la DSE e' es de tipo **entry** (inserción) o **glue** (adhesivo), añadir el RDN de e' a **listInfo.subordinates** con **aliasEntry** fijado a **FALSE** y **fromEntry** fijado según que e' sea o no una copia.
 

NOTA – En el caso de que e' sea **glue** (adhesivo), debe tener uno o más subordinados, lo que implica que no puede ser un alias del DSA maestro. Asimismo, cualquier ACI relevante para Listar se almacena en esta DSE, proporcionada mediante el protocolo de sombreado.
  - iii) Si la DSE e' es de tipo **alias**, añadir el RDN de e' a **listInfo.subordinates** con **aliasEntry** fijado a **TRUE**, y **fromEntry** fijado según que e' sea o no una copia.
- c) Comprobar si se ha rebasado el límite de tiempo, tamaño o administrativo. En este caso, fijar correspondientemente **limitProblem** en **partialOutcomeQualifier** y retornar.
- d) Continuar desde el paso 3) a) hasta que se hayan procesado todas las DSE subordinadas.
- 4) Si se han procesado todas las DSE subordinadas, retornar al **despachador de operaciones**.
- 5) Para cada subinserción e' inmediatamente subordinada a DSE e ejecutar los siguientes pasos:
  - a) Comprobar la ACI en e'. Si la ACI desautoriza el listado de RDN de e', saltar esta DSE. De no ser así, añádase el RDN de e' a **listInfo.subordinates** con **aliasEntry** fijado a **FALSE** y **fromEntry** fijado según que e' sea o no una copia.
  - b) Comprobar si se ha rebasado el límite de tiempo, de tamaño o administrativo. En este caso, fijar correspondientemente **limitProblem** en **partialOutcomeQualifier** y retornar.
- 6) Retornar al **despachador de operaciones**.

### 19.3.1.2.2 Procedimiento List (II) [listar (II)]

El **procedimiento List (II)** se compone de los siguientes pasos, representados en la figura 18:

- 1) Para cada DSE e' inmediatamente subordinada a DSE e, ejecutar los pasos 1) a) a 1) d):
  - a) Comprobar ACI en e'. Si la operación está desautorizada por la ACI, continuar con la siguiente subordinado inmediato de e.
  - b) Añadir el RDN de DSE e' a **listInfo.subordinates**, con el componente **aliasEntry** de **listInfo.subordinates** fijado según que e' sea o no un alias, y el componente **fromEntry** fijado según que e' sea o no una copia. Ignorar los DSE de tipo **shadow**, si **excludeShadows** es **TRUE**.
  - c) Comprobar si se ha superado el límite de tiempo, tamaño o administrativo. Si es así, fijar correspondientemente el **limitProblem** de **partialOutcomeQualifier** y retornar.
  - d) Continuar desde el paso 1) a) hasta que se hayan procesado todas las DSE subordinadas.



- 2) Si se han procesado todas las DSE subordinadas, comprobar si esta subpetición viene de DAP o de DSP. En el caso de que esta subpetición haya sido sometida vía DAP y el **ListResult** esté vacío, retornar un **serviceError** con problema **invalidReference** al despachador de operaciones. En otro caso, retornar el **ListResult**.

NOTA – **invalidReference** se utiliza como una medida de precaución, por razones de seguridad, en el caso en que el usuario no tenga acceso a la inserción superior. Si está disponible la ACI de la inserción superior (proporcionada por la RMOB), puede retornarse un resultado nulo si se permite.

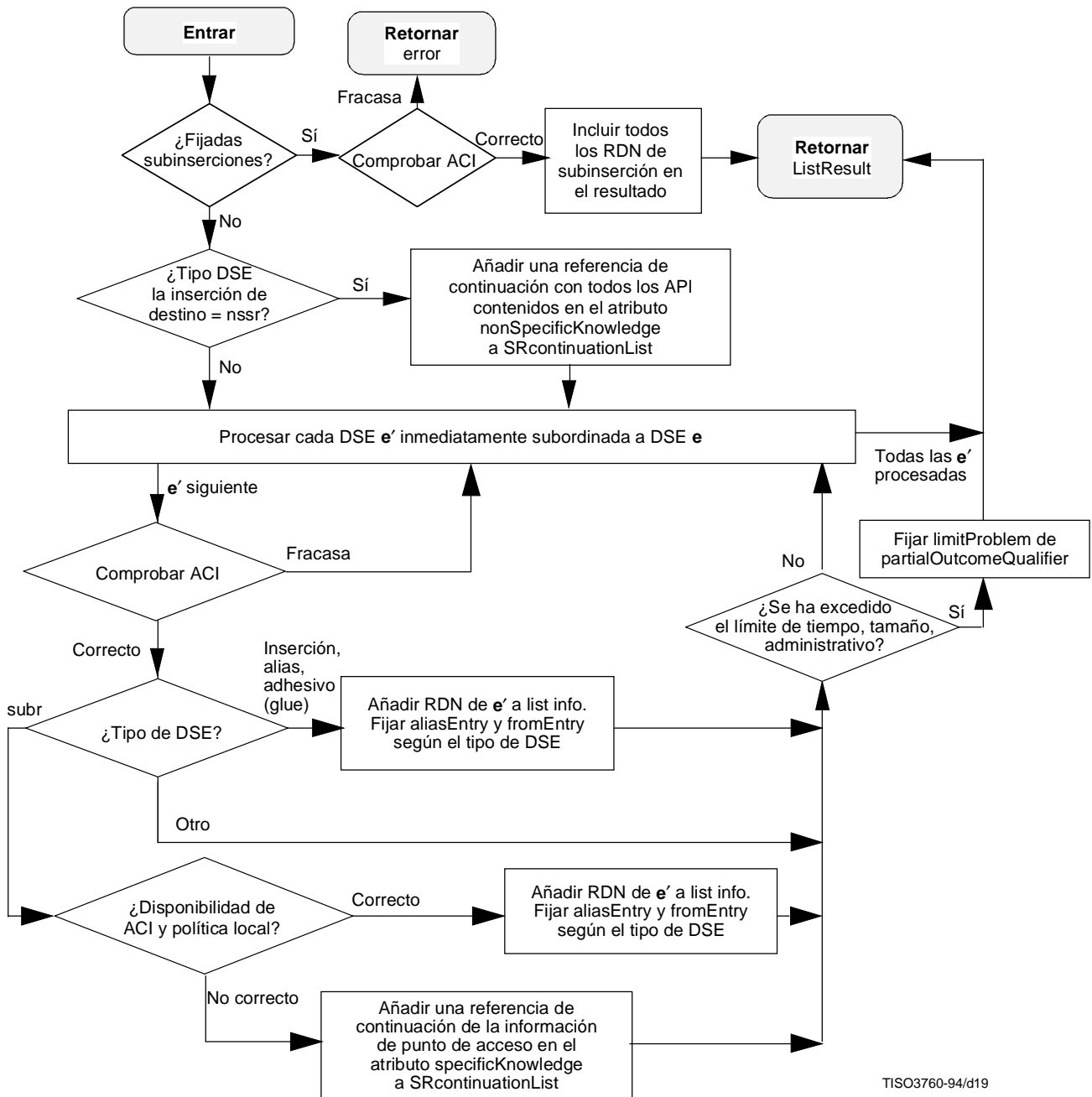


Figura 17 – Procedimiento List (I) [listar (I)]

### 19.3.2 Procedimiento Search (buscar)

Esta subcláusula especifica el procedimiento de evaluación específico a las operaciones **search** y **chainedSearch**.

El procedimiento **Search (I)** se seguirá cuando el componente **operationProgress.nameResolutionPhase** de la petición de **búsqueda** esté fijado a **notStarted** o **proceeding** y cuando el DSA, tras realizar la resolución de nombre, encuentra

que tiene el objeto de base. El procedimiento **Search (II)** se seguirá cuando el componente **nameResolutionPhase** de la petición de búsqueda está fijado a **complete**.

NOTA – Cuando la **nameResolutionPhase** es **completed**, cabe esperar que el objeto de destino sea el superior inmediato de un prefijo de contexto.

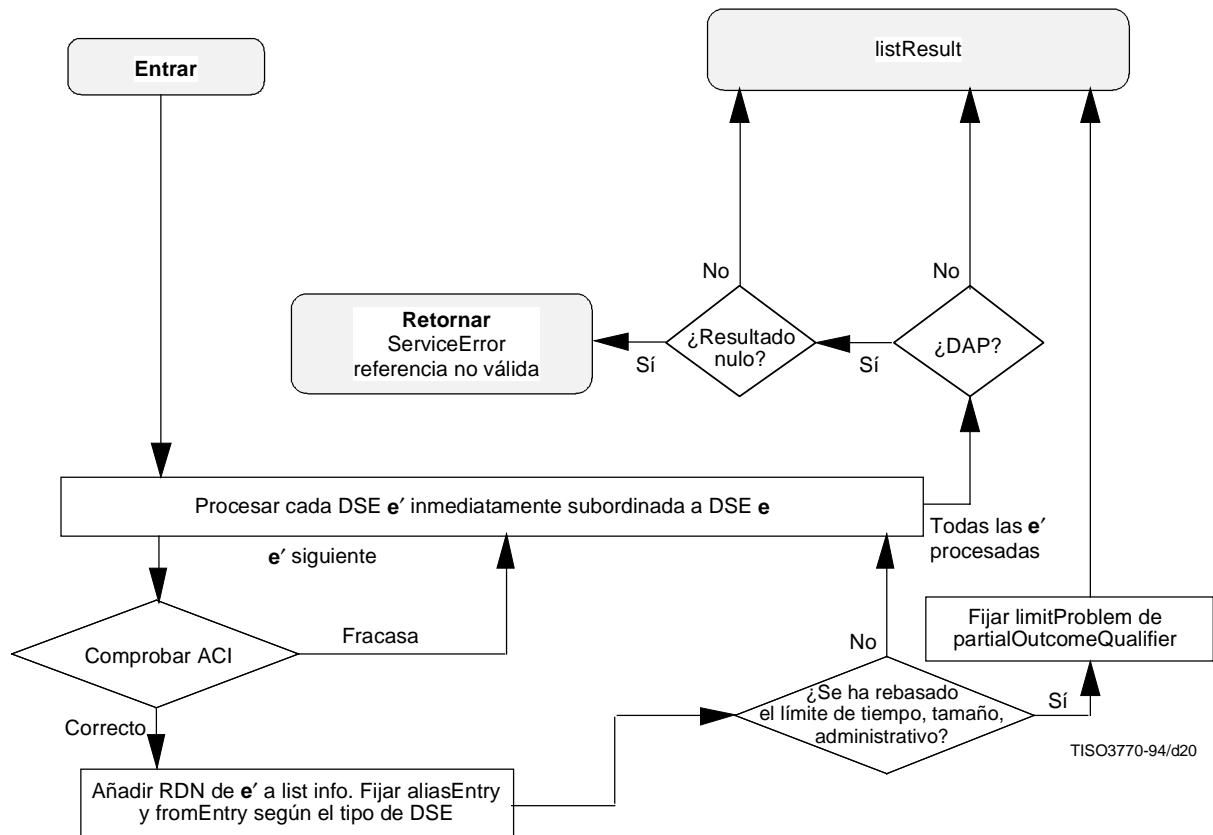


Figura 18 – Procedimiento List (II) [listar (II)]

### 19.3.2.1 Parámetros del procedimiento

#### 19.3.2.1.1 Argumentos

Los argumentos utilizados por este procedimientos son:

- el **SearchArgument**;
- la DSE **e** de destino;
- **operationProgress** del **ChainingArguments**;
- **exclusions** del **ChainingArguments** (una lista de RDN a excluir de la búsqueda).

#### 19.3.2.1.2 Resultados

Si este procedimiento se ejecuta con éxito, retorna:

- un conjunto de inserciones concordadas en **searchResult.entryInformation**;
- **alreadySearched** en **ChainingResults**;
- un conjunto de referencias de continuación en **SRcontinuationList**.

### 19.3.2.2 Definición de procedimientos

#### 19.3.2.2.1 Procedimiento Search (I) [buscar (I)]

Este es un procedimiento recursivo que se aplica a una petición de **búsqueda** que comienza en una inserción de destino **e** dada. El procedimiento busca la inserción de destino **e** y luego procesa las DSE inmediatamente subordinadas a **e**. El procedimiento se invoca por sí mismo recursivamente en el caso de que la búsqueda comprenda todo un subárbol. El procedimiento se compone de los siguientes pasos según se muestra en la figura 19:

- 1) Si el tipo de DSE **e** es **cp** (una DSE en un prefijo de contexto), comprobar si cualquier elemento del argumento **exclusions** es un prefijo del DN de **e**.
  - a) Si es así, retornar.
  - b) Si no, llamar el procedimiento comprobar idoneidad.
    - i) Si **e** es inidónea, hacer una continuation Reference como sigue y añadirla a la **SRcontinuationList**:
      - **targetObject** al nombre distinguido primario del inmediato superior del DSE **e** (en los RDN pueden incluirse valores distinguidos alternativos);
      - **aliasedRDNs** ausente;
      - **operationProgress** con **nameResolutionPhase** fijado a **completed** y **nextRDNTobeResolved** ausente;
      - **rdnsResolved** ausente;
      - **referenceType** fijado a **supplier**;
      - **accessPoints** fijado al **AccessPointInformation** derivado del (de los) valor(es) hallado(s) a partir del atributo **supplierKnowledge** en **e**.

A continuación, retornar.

NOTA 1 – Este es el único lugar en que una subpetición de búsqueda (**nameResolutionPhase** está **completed**) es concatenada a un proveedor de sombra. En otras palabras, el objeto de destino para esta subpetición concatenada es siempre un prefijo de contexto.

- ii) En otro caso, añadir el DistinguishedName de **e** a **alreadySearched** en **ChainingArguments**.

NOTA 2 – **alreadySearched** sólo contiene prefijos de contexto.

- 2) Si **e** es del tipo **alias** y **searchAliases** en **SearchArgument** es **TRUE** llamar el procedimiento **Search Alias** y retornar.
- 3) Si **subset** es **oneLevel**, continuar en el paso 6).
 

NOTA 3 – La **e** no puede ser subordinado incompleto en este punto, ya que el procedimiento comprobar idoneidad en el prefijo de contexto deberá haber determinado que esto no puede suceder.
- 4) Si **subset** es **baseObject**, o si **entryOnly** es **True**, continuar en este paso. De no ser así continuar en el paso 5).

Si uno de los siguientes es True:

- a) **e** es de tipo **subentry** y está fijado el control de servicio **subentry**; o
- b) **e** no es de tipo **subentry** y no está fijado el control de servicio **subentry**;

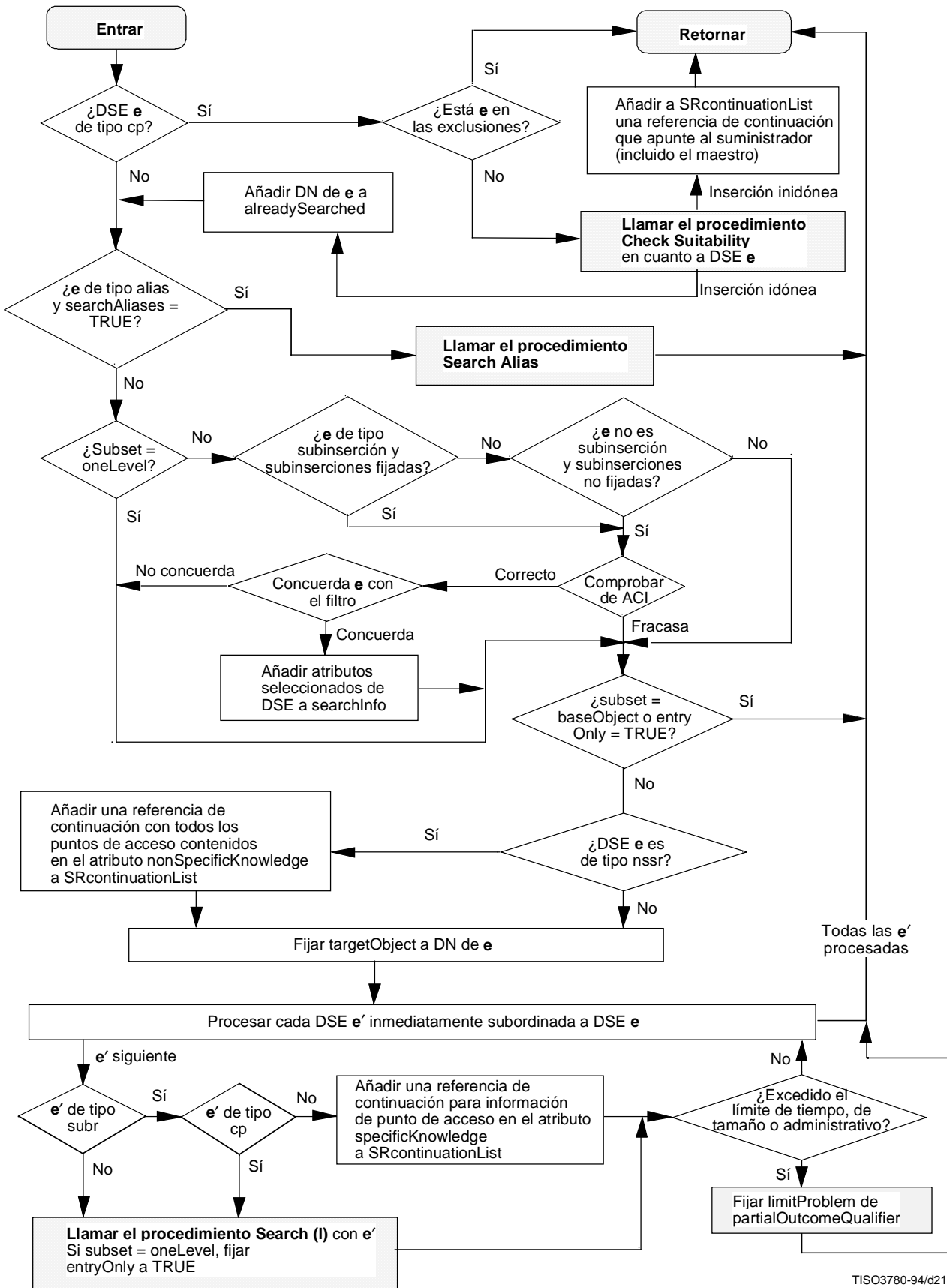
efectuar entonces los siguientes pasos:

- i) Comprobar la ACI. Si la operación está desautorizada, retornar.
- ii) Aplicar el argumento de filtro especificado en el **SearchArgument.filter** a la DSE **e**. Asegurarse de que el acceso a todos los atributos utilizados en el filtro está permitido según se define en la Rec. UIT-T X.501 | ISO/CEI 9594-2. Si el filtro concuerda, añadir los atributos seleccionados por **SearchArgument.selection** a la lista de inserciones concordados en **searchResult.entryInformation**. Añadir sólo atributos que no sean mayores que el **attributeSizeLimit**.

Retornar.

- 5) Si **subset** es **subtree** (y **entryOnly** no es **TRUE**), y, además, una de las situaciones siguientes es **TRUE**:
- e** es de tipo **subentry** y está fijado el control de servicio **subentry**; o
  - e** no es de tipo **subentry** y no está fijado el control de servicio **subentry**;  
efectuar los siguientes pasos:
    - Comprobar **ACI**. Si la operación está desautorizada, continuar en el paso 6).
    - Aplicar el argumento de filtro especificado en el **SearchArgument.filter** a la DSE **e**. Asegurar que se permite el acceso a todos los atributos utilizados en el filtro según se define en la Rec. UIT-T X.501 | ISO/CEI 9594-2. Si el filtro concuerda, añadir los atributos seleccionados **SearchArgument.selection** a la lista de inserciones concordados de **searchResult.entryInformation**.
    - Avanzar al paso 6).
- 6) Si **e** es del tipo **nssr**, añadir una continuation Reference a **SRcontinuationList** con los siguientes componentes:
- **targetObject** al nombre distinguido primario del DSE **e** (en los RDN pueden incluirse valores distinguidos alternativos);
  - **aliasedRDNs** ausente;
  - **operationProgress** con **nameResolutionPhase** fijado a **completed** y **nextRDNtoBeResolved** ausente;
  - **rdnsResolved** ausente;
  - **referenceType** fijado a **nssr**;
  - **accessPoints** fijado a **AccessPointInformation** derivado del (de los) valor(es) hallado(s) en el atributo **nonSpecificKnowledge**.
- 7) Procesar todas las DSE **e'** localizadas inmediatamente subordinadas a la DSE **e** de destino hasta que se hayan procesado todas las DSE subordinadas. Durante este bucle, si la lista de los inserciones concordados en **searchResult.entryInformation** rebasa el límite de tamaño, de tiempo o administrativo, fijar correspondientemente **limitProblem** en **partialOutcomeQualifier** y retornar.
- NOTA 4 – La comprobación del límite de tamaño también se aplica implícitamente cada vez que se actualiza **searchResult**.
- Si la DSE **e'** es de tipo **subr**, y no de tipo **cp**, añadir una continuation Reference a **SRcontinuationList** con los siguientes componentes:
    - **targetObject** al nombre distinguido primario del DSE **e** (en los RDN pueden incluirse valores distinguidos alternativos);
    - **aliasedRDNs** ausente;
    - **operationProgress** con **nameResolutionPhase** fijado a **completed** y **nextRDNtoBeResolved** ausente;
    - **rdnsResolved** ausente;
    - **referenceType** fijado a **subr**;
    - **accessPoints** fijado a la información de punto de acceso contenida en el atributo **specificKnowledge** de DSE **e'**.

NOTA 5 – Si **e'** es de tipo **cp** y de tipo **subr**, es posible generar una subpetición de búsqueda o desde la referencia subordinada o desde el conocimiento de proveedor, pero no desde ambos. Este procedimiento utiliza el último caso (referencias de proveedor halladas en **cp**).
  - Para todos los casos:
    - Si el parámetro **subset** es **oneLevel**, fijar **entryOnly** a **TRUE**.
    - Ejecutar recursivamente el procedimiento **Search (I)** para la DSE **e'** de destino.
- 8) Si se han procesado todos los subordinados, retornar al despachador de operaciones para continuar con el procesamiento.



TISO3780-94/d21

Figura 19 – Procedimiento Search (I) [(buscar (I))]

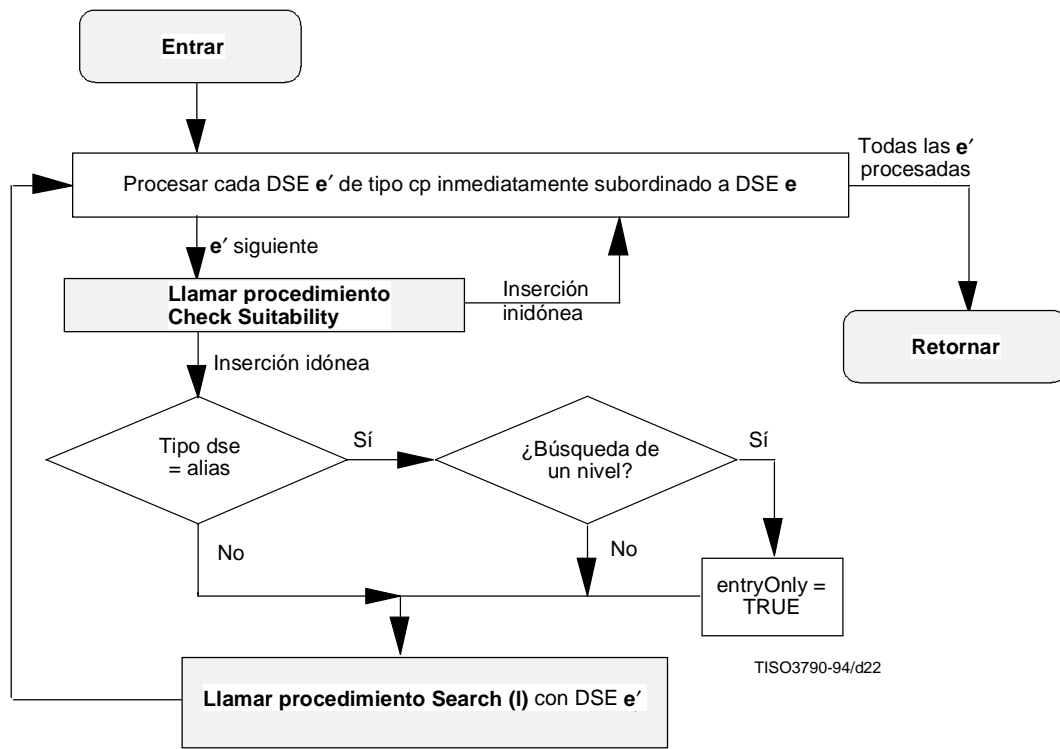


Figura 20 – Procedimiento Search (II) [buscar (II)]

#### 19.3.2.2.2 Procedimiento Search (II) [buscar (II)]

Este procedimiento es aplicable si se procesa una petición de búsqueda procedente de una descomposición de petición en el DSA desde el que se recibió la petición. El procedimiento procesa las DSE situadas debajo de la DSE *e* de destino y llama el procedimiento **Search (I)** para cada inserción de objeto:

- 1) Procesar todas las DSE *e'* que están inmediatamente subordinadas a la DSE *e* de destino hasta que hayan sido procesadas todas las DSE subordinadas. Una vez procesadas todas las subordinadas, retornar al despachador de operaciones para continuar el procesamiento.
- 2) Si la DSE no es de tipo **cp**, ignorarla y volver al paso 1).
- 3) Llamar el procedimiento **Check Suitability** (comprobar idoneidad). Si es idónea, continuar en el paso 4); si no lo es, ignorarla y volver al paso 1).
- 4) Ejecutar el **procedimiento de búsqueda (I)** con respecto a la DSE *e'* como se describe en 19.3.2.2. Si la DSE es de tipo **alias** y el valor del parámetro **subset** está fijado a **oneLevel**, fijar **ChainingArguments.entryOnly** a **TRUE** cuando se llame el procedimiento de **búsqueda (I)**. Volver al paso 1).

#### 19.3.2.2.3 Procedimiento Search Alias (buscar alias)

Se ejecuta este procedimiento si se ha encontrado una DSE de tipo **alias** durante el procesamiento de una petición de búsqueda.

- 1) Si **subset** es **baseObjet** o **oneLevel**, continuar en el paso 4).
- 2) Si **aliasedEntryName** es un prefijo de **targetObjet** o **baseObject** o cualesquiera de los valores previos del objeto de destino en **ChainingArguments.tracelInformation**, el alias se excluye de la búsqueda porque esto produciría una búsqueda recursiva con resultados duplicados.
- 3) Si **targetObjet** o **baseObject** o cualesquiera de los valores previos del **targetObject** en **ChainingArguments.tracelInformation**, es un prefijo de **aliasedEntryName**, no se requiere procesamiento específico del alias porque el subárbol con alias será buscado de todos modos.

NOTA – Para los tres casos anteriores, **baseObject** puede no ser prefijo de **targetObject**, debido al desreferenciado de alias.

- 4) Construir una petición **DSP** local con el objeto de destino fijado al **aliasedEntryName**. Si **subset** es **oneLevel**, fijar **entryOnly** a **TRUE**. Llamar al despachador de operaciones para continuar localmente la petición.
  - 5) Si el despachador de operaciones retorna un error de referimientto, o errores de ocupado o indisponible, añadir (o generar y añadir) la referencia de continuación a **partialOutcomeQualifier** de **SearchResult**, y retornar.
  - 6) Si el despachador de operaciones retorna otros errores, descartarlo y retornar.
  - 7) Si el despachador de operaciones retorna un **SearchResult**, entonces:
    - i) Si el resultado está firmado, criptado o firmado y criptado, añadirlo a **uncorrelatedSearchInfo** en **SearchResult**.
    - ii) Si el resultado no está firmado, criptado o firmado y criptado, añadirlo a **searchInfo** en **SearchResult**.
- Y retornar.

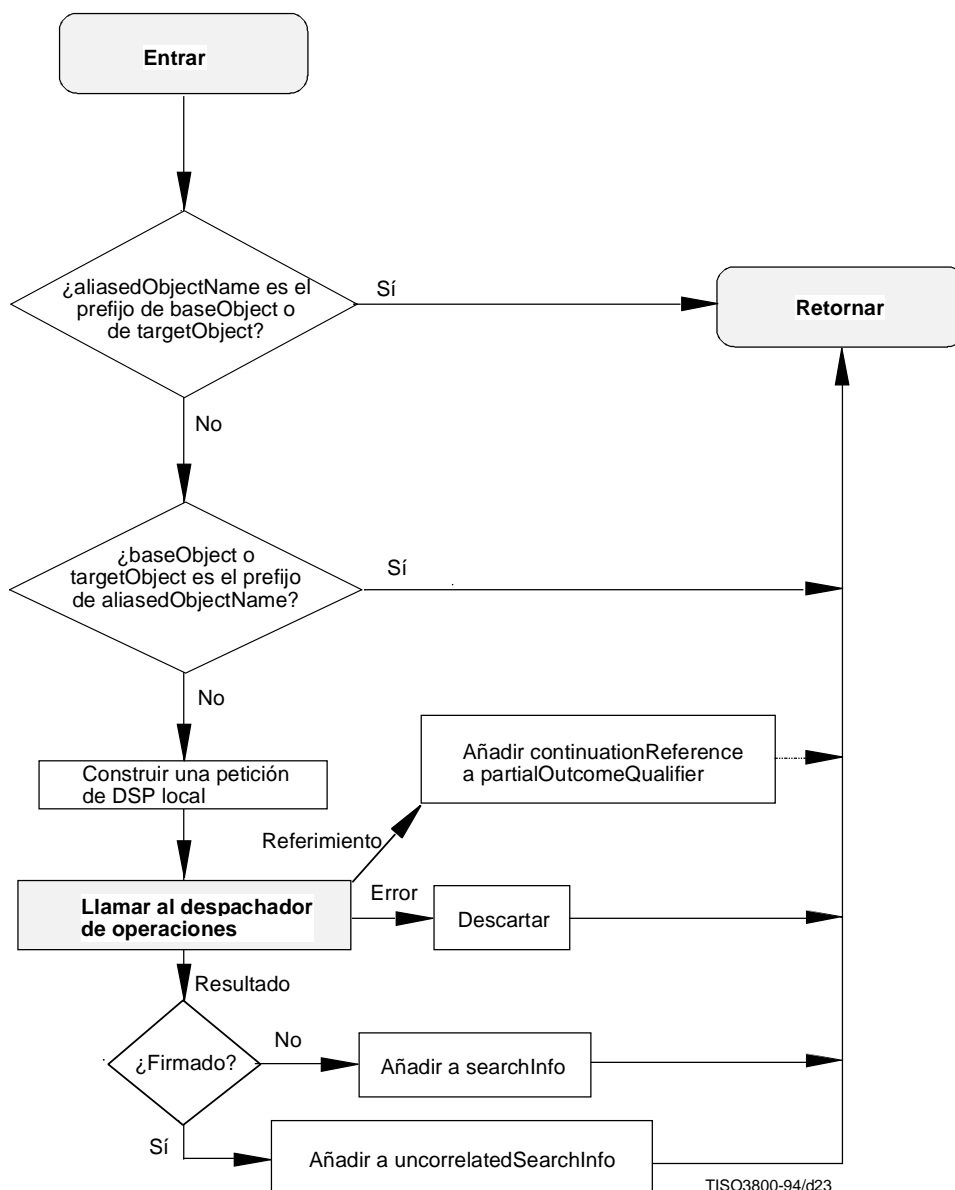


Figura 21 – Procedimiento Search alias (buscar alias)

## 20 Procedimientos de referencia de continuación

Los procedimientos de esta cláusula se llaman para procesar la lista de referencias de continuación (**NRcontinuationList** o **SRcontinuationList**) creada por otros procedimientos.

Los procedimientos de **referencia de continuación** se componen de los pasos mostrados en las figuras 24, 25 y 26. La primera etapa consiste en identificar conjuntos de referencias de continuación, de la lista de continuación, que tienen un componente de objeto de destino común. Estos se han creado desde un conjunto de referencias subordinadas o subordinadas no específicas asociadas a la misma inserción en el DIT. Dentro de cada uno de estos conjuntos puede haber referencias de continuación que tienen lugar más de una vez. Estos conjuntos se deben explorar y descartar cualquier duplicado.

El DSA puede procesar estos conjuntos (cada uno con un componente **targetObject** diferente) independientemente, bien en serie o en paralelo, pues no hay riesgo de que se retornen los mismos resultados desde dos conjuntos cualesquiera. Sin embargo, el procesamiento de cada referencia de continuación en un conjunto, de cada **AccesPointInformation** en una referencia de continuación, de cada punto de acceso en una **AccesPointInformation**, ha de ser controlado, o de lo contrario pueden aparecer resultados duplicados, como se describe en 20.1.

El procedimiento adoptado en el procedimiento **APInfo** consiste en procesar uno por uno el conjunto de puntos de acceso contenidos en una **AccesPointInformation** único. Todos éstos apuntan a (copias de) un mismo contexto de denominación (o quizás a un conjunto de contextos de denominación en el caso de NSSR). Si el primer punto de acceso produce un resultado o un error duro, no tienen que continuar procesándose los puntos de acceso. Sin embargo si el error es un error blando, es decir, un **serviceError** (**busy**, **unavailable**, **unwillingToPerform**, **invalidReference** o **administrativeLimitExceded**), el DSA puede optar, como opción local, por procesar otro punto de acceso del conjunto.

El procesamiento de los valores de **AccesPointInformation** en un conjunto de referencias de continuación se maneja de manera uniforme, independientemente de la referencia de continuación desde la que se originaron. [Esto se debe a que dos DSE de tipo **subr** por debajo de una única inserción producirían dos referencias de continuación, cada una de las cuales con un valor **AccessPointInformation**, mientras que una DSE de tipo **nssr** a los mismos dos subordinados (suponiendo que se encuentran en DSA diferentes) producirían una referencia de continuación que contendría un conjunto de dos valores **AccessPointInformation**.]

Los valores de **AccessPointInformation** pueden procesarse bien en serie o en paralelo, como se describe en 20.1. La estrategia paralelo tiene más probabilidades de producir resultados duplicados. Los duplicados se descartarán siempre.

### 20.1 Estrategia de concatenación en presencia de sombreado

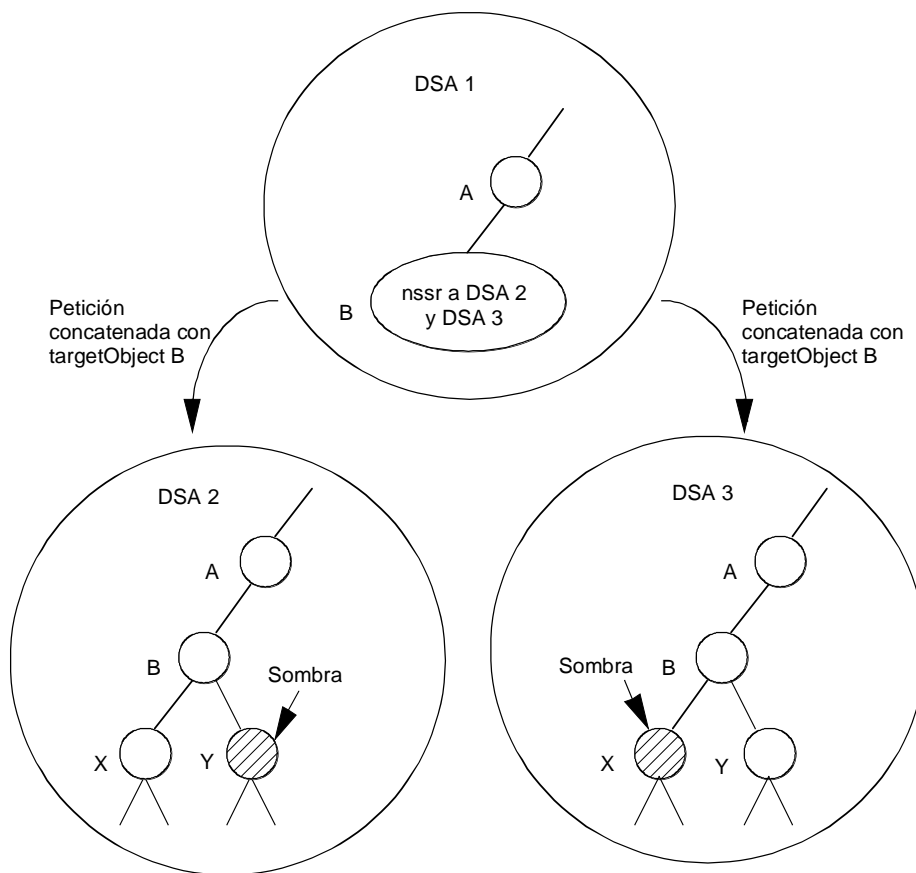
En presencia de sombreado, un DSA puede elegir entre diferentes estrategias cuando tiene que multiconcatenar la petición, es decir, concatenarla a más de un DSA. Esta opción siempre se da si el DSA tiene que procesar más de una referencia de continuación con el mismo **targetObject**. Esta situación puede provenir de la multiconcatenación causada por descomposición de NSSR durante la resolución de nombre (como se muestra en la figura 22), o de la descomposición de petición durante la evaluación de una operación de objeto múltiple (véase la figura 23).

El objetivo de estas estrategias es afrontar el problema de los resultados duplicados y del procesamiento duplicado cuando se utiliza información sombreada en la multiconcatenación de peticiones (causada o por descomposición de NSSR o de petición). Por ejemplo, en la figura 22, DSA 1 multiconcatena una petición a los DSA 2 y 3 debido a la NSSR contenida en la DSE B. Si se permite el uso de información sombreada, DSA 2 y 3 pueden aplicar la operación concatenada a ambos subárboles comenzando en X e Y.

De forma similar, en la figura 23, DSA 1 multiconcatena (como resultado de la descomposición de petición) a las dos referencias subordinadas contenidas en X e Y de las DSE. Una vez más, si se permite el uso de información sombreada, los DSA 2 y 3 pueden aplicar la operación concatenada a ambos subárboles comenzando en X e Y.

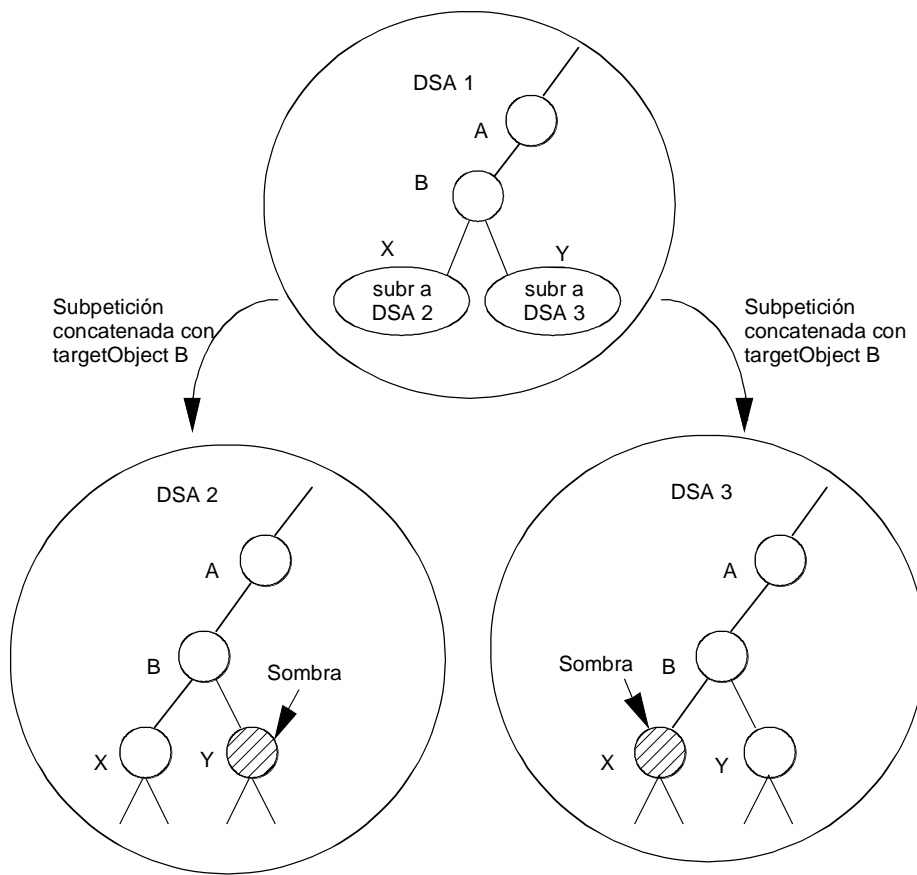
Para afrontar este problema de duplicación un DSA puede optar por una de las siguientes estrategias cuando se multiconcatena a múltiples DSA peticiones con el mismo **targetObject**.





TISO3810-94/d24

**Figura 22 – Multiconcatenación causada por descomposición de NSSR durante una resolución de nombre**



TISO3820-94/d25

**Figura 23 – Multiconcatenación causada por descomposición de petición mediante referencias subordinadas**

### 20.1.1 Estrategia sólo maestro (master only)

Un DSA puede elegir esta estrategia para evitar el uso de información sombreada cuando realiza una multiconcatenación paralela o secuencial causada por la descomposición de NSSR o la descomposición de una petición durante una evaluación de buscar o listar. Para esta estrategia, durante la evaluación de una operación de buscar o listar, el componente **excludeShadows** del **ChainingArguments** se fija a **TRUE**. Si durante una resolución de nombre se encuentran NSSR, un DSA puede fijar **nameResolveOnMaster** a **TRUE** para asegurar que habrá de seguirse un solo trayecto. **nameResolveOnMaster** se fijará **TRUE**, si se encuentran NSSR y la operación es una de las operaciones de modificación de directorio. Tanto en uno como en otro caso, solamente el DSA o los DSA que tienen la inserción o inserciones maestras relevantes para la operación realizarán dicha operación. Esta estrategia de sólo maestro puede utilizarse tanto en la multiconcatenación paralelo como en la secuencial.

NOTA – La fijación de **nameResolveOnMaster** a **TRUE** elimina la posibilidad de múltiples trayectos durante la resolución de nombre por 1) ignorando entradas de sombra y 2) asegurando que sólo un DSA puede continuar con resolución de nombre en situaciones en las que una distribución DIT compleja podría permitir continuar a más de uno. Esto se lleva a cabo permitiendo que sólo el DSA que tiene la inserción maestra correspondiente a los primeros RDN **nextRDNTToBeResolved** del nombre de objeto de destino continúe con la resolución de nombre. Cualquier otro DSA no podrá continuar aun cuando pueda tener inserciones maestras que concuerdan más con el nombre de objeto de destino.

### 20.1.2 La estrategia paralelo

Cuando se utiliza esta estrategia, un DSA envía todas las peticiones concatenadas mediante multiconcatenación en paralelo. Esta estrategia puede utilizarse durante la evaluación de buscar o listar, y una resolución de nombre de NSSR. Esto permitirá el uso de información sombreada para procesar las peticiones concatenadas, pero puede dar lugar a ejecuciones duplicadas y a resultados duplicados de la operación. Si un DSA elige esta estrategia, eliminará los resultados duplicados del resultado de la operación que retorna.

Como no es posible la eliminación de resultados duplicados si se ha solicitado un resultado firmado, un DSA no elegirá esta estrategia si se solicitan resultados firmados durante la evaluación de buscar, a menos que también esté fijado **excludeShadows**.

### 20.1.3 Estrategia secuencial

Esta estrategia evita resultados duplicados mediante el empleo de una multiconcatenación secuencial para procesar las (sub)peticiones concatenadas de una descomposición de buscar o de una descomposición de NSSR. Todas las peticiones concatenadas son procesadas, una tras otra.

En el caso de descomposición de NSSR, si se retorna un error fuerte con respecto a una petición, las peticiones posteriores no tienen que ser concatenadas. Si se retorna un error blando, se podrá concatenar una petición ulterior, o retornar al peticionario un error blando, lo que dependerá de la política local que se siga.

En el caso de evaluación de buscar, el componente **exclusion** de los **ChainingArguments** se fija al conjunto de RDN que ya ha sido procesado. Esto se hace incorporando los elementos en **ChainingResults.alreadySearched** al argumento **exclusions** de la siguiente petición concatenada. Esta es la única estrategia que evita totalmente las duplicaciones durante una evaluación de buscar.

No está definida una estrategia secuencial para evaluación de listar (aunque puede utilizarse multiconcatenación secuencial), pues un DSA superior no tiene medios de impedir que subordinados específicos sean retornados en ulteriores subpeticiones de listar (obsérvese que **excludeShadows** no excluye subordinados específicos, sino que más bien es una manera bruta de excluir todas las sombras).

## 20.2 Envío de subpeticiones concatenadas a un DSA distante

Antes de enviar una subpetición, un DSA tiene que ejecutar una operación **dSABind** (**vincular**) cuando el DSA en cuestión tiene que establecer una asociación con el DSA distante. La gestión de las asociaciones está fuera del alcance de estas Especificaciones de directorio. Una asociación con otro DSA se considera no disponible si no puede ser establecida, o si el DSA decide, por razones locales, no establecerla. En este caso ha fracasado **dSABind**. Es una decisión local la relativa al momento de dejar de intentar establecer una asociación y declararla no disponible.

Cuando un DSA intenta **dSABind** a otro DSA y recibe un **directoryBindError**, ha fracasado el envío de la subpetición.

## 20.3 Parámetros del procedimiento

### 20.3.1 Argumentos

Este procedimiento hace uso de los siguientes argumentos:

- La lista de referencias de continuación a procesar en **NRcontinuationList** (para el procedimiento de **referencia de continuación de la resolución de nombre**) y de **SRcontinuationList** (para los procedimientos de **referencia de continuación de listar y buscar**, respectivamente).
- Los **CommonArguments** del argumento de la operación.
- El **ChainingArguments**.

### 20.3.2 Resultados

Estos procedimientos crean los siguientes resultados:

- Una lista de resultados/errores recibidos de peticiones concatenadas enviadas si se ha seleccionado la concatenación.
- Una lista actualizada de referencias de continuación no procesadas en **continuationList**.

### 20.3.3 Errores

Estos procedimientos pueden retornar uno de los siguientes errores:

- Un **serviceError outOfScope** en el caso de que se hubiese creado un referimiento que no está dentro de **scopeOfReferral**.
- Un **serviceError ditError** en el caso de que se haya detectado una referencia de conocimiento no válida.
- Un **nameError noSuchObject** en el caso de que todas las subpeticiones provenientes de la descomposición de NSSR retornaron **unableToProceed**.
- Cualquier otro error retornado por una subpetición concatenada.
- Un **referral** en el caso de que no se seleccionó la concatenación y **operationProgress.nameResolutionPhase** esta fijada a **notStarted** o **proceeding**.

## 20.4 Definición de los procedimientos

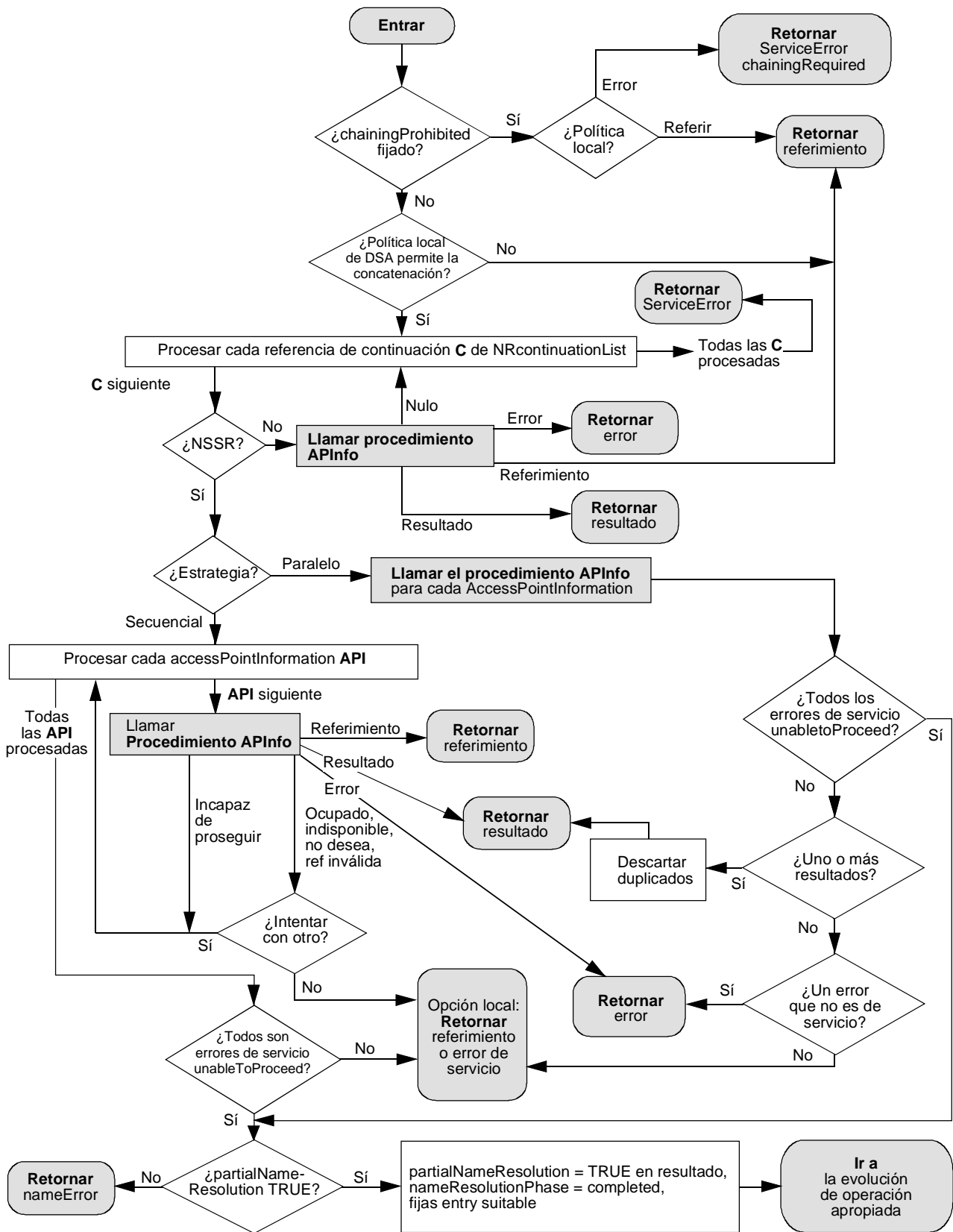
Si **operationProgress.nameResolutionPhase** está fijada a **notStarted** o a **proceeding**, se seguirá el procedimiento de 20.4.1 (procedimiento de **referencia de continuación de resolución de nombre**). Las operaciones de interrogación de objeto múltiple List (listar) y Search (buscar) llaman respectivamente los procedimientos indicados en 20.4.2 y 20.4.3.

### 20.4.1 Procedimiento Name Resolution Continuation Reference (referencia de continuación de resolución de nombre)

El procedimiento de **referencia de continuación de resolución de nombre** se compone de los pasos descritos en la figura 24. El principio básico de este procedimiento consiste en procesar secuencialmente el conjunto de referencias de continuación creadas durante una resolución de nombre. Se ejecutarán los siguientes pasos para cada referencia de continuación **C** contenida en **NRcontinuationList** en un orden seleccionado, hasta que se hayan procesado todas las referencias o se haya retornado un error o resultado. Si se han procesado todas las referencias, retornar al **despachador de operaciones** para proseguir con el procedimiento de **fusión de resultados** para procesar el resultado o referimiento recibidos.

- 1) Comprobar si está fijado **chainingProhibited**. Si lo está, no se permite al DSA concatenar. De acuerdo con **local policy** (política local), se retorna un **serviceError** con el problema **chainingRequired** o se retorna un **referimiento** al **despachador de operaciones**.
- 2) Si no está fijado **chainingProhibited**, comprobar si **local policy** permite concatenar. Si no se permite concatenar, retornar un **referimiento**. Si **local policy** permite concatenar, avanzar al siguiente paso.

- 3) Procesar cada una de las referencias de continuación de la lista de referencias de continuación halladas en **NRcontinuationList**. Si no hay más referencias de continuación no procesadas, retornar con **serviceError**.
- 4) Procesar la siguiente **C** de referencias de continuación de la **NRcontinuationList**. Si es una NSSR, continuar en el paso 5). Si no es una NSSR, llamar el procedimiento **APIInfo** para procesarla. Actuar en función de lo que pueda retornar el procedimiento **APIInfo**:
  - Si el procedimiento **APIInfo** retorna un **null result**, continuar en el paso 3) procesando la siguiente referencia de continuación.
  - Si el procedimiento **APIInfo** retorna un **error**, **referral** o **result**, retornarlo.
- 5) En este caso, la referencia de continuación es de tipo NSSR y el DSA tiene la opción de realizar una concatenación secuencial o en paralela, en función de la **elección local de estrategia**. Si la NSSR va a procesarse secuencialmente, continuar en el paso 6). Si va a procesarse en paralelo, para cada uno de las **AccesPointInformation (API)** en la NSSR se llama el procedimiento **APIInfo** para que sean procesadas en paralelo. Esperar a que se procesen todas las API, es decir esperar a que todas las llamadas al procedimiento **APIInfo** hayan retornado. Comprobar todos los resultados recibidos de las llamadas al procedimiento **APIInfo** en el siguiente orden:
  - Si la llamada retorna un **serviceError** con el problema **unableToProceed** y **partialNameResolution** es **FALSE**, retornar **nameError**.
  - Si la llamada retorna un **serviceError** con el problema **unableToProceed** y **partialNameResolution** es **TRUE**, en el resultado fijar **partialName** a **TRUE**, **nameResolutionPhase** a **completed**, fijar **entry suitable** (para la **lastEntryFound**), e ir a la evaluación de operación apropiada.
  - Si se recibe uno o más **resultados**, **descartar los posibles duplicados** y retornar el **resultado**.
  - Si se recibe un **error** que no es un **serviceError** (por ejemplo un **nameError**), retornar un **error**.
  - En otro caso, retornar un **referimiento** o **serviceError** al **despachador de operaciones**, según la opción local.
- 6) Elegir la siguiente **API** no procesada del conjunto de **API** en la NSSR y continuar en el paso 7). Si se han procesado todas las **API**, comprobar si todas las llamadas al procedimiento **APIInfo** retornaron un **serviceError** con el problema **unableToProceed**.
  - Si así fue y **partialNameResolution** es **FALSE**, no se puede encontrar la inserción y se retorna un **nameError**. Si así fue y **partialNameResolution** es **TRUE**, en el resultado poner **partialName** a **TRUE**, **nameResolutionPhase** a **completed**, fijar **entry suitable** (para la **lastEntryFound**), e ir a la evaluación de operación apropiada. Si no fue así, retornar, según la opción local, un **referral** o un **serviceError**.
- 7) Llamar el procedimiento **APIInfo**. Actuar según los posibles resultados de la llamada al procedimiento **APIInfo**:
  - Si se recibe un **serviceError** con el problema **unableToProceed**, intentar otro punto de acceso. Continuar en el paso 6).
  - Si se recibe un **serviceError** con el problema **busy**, **unavailable**, **unwillingToPerform** o **invalidReference**, el problema indicado puede ser de carácter transitorio y es una opción local tratar de concatenar la petición a otro DSA. Si se opta por probar con otro DSA, continuar en el paso 6). En caso contrario, retornar un **referral** o **serviceError**, según la decisión local.
  - Si se recibe un error distinto a **serviceError** con el problema **busy**, **unavailable**, **unwillingToPerform**, **invalidReference** o **unableToProceed**, ese error debe retornarse al **despachador de operaciones**. Si el **serviceError** es **invalidReference**, ésta deberá convertirse en **ditError** antes de retornarla al peticionario.
  - Si se recibe un **result** o **referral**, retornarlo al **despachador de operaciones**.



TISO8980-99/d26

Figura 24 – Procedimiento Name Resolution Continuation Reference (referencia de continuación de resolución de nombre)

20.4.2 Procedimiento List Continuation Reference (listar referencia de continuación)

El procedimiento **listar referencia de continuación** consta de los pasos mostrados en la figura 25. Se invoca este procedimiento cuando una petición de listar no puede satisfacerse en el DSA local y se ha añadido un conjunto de referencias de continuación a **SRcontinuationList** para concatenación o referimiento. Todas estas referencias de continuación (CR) tienen el mismo **targetObjet**. Las CR con **referenceType nssr** tienen uno o más valores **AccessPointInformation** (API), mientras que CR de otro tienen sólo una API. Cada una de estas API se extrae y se considera para concatenación o referimiento.

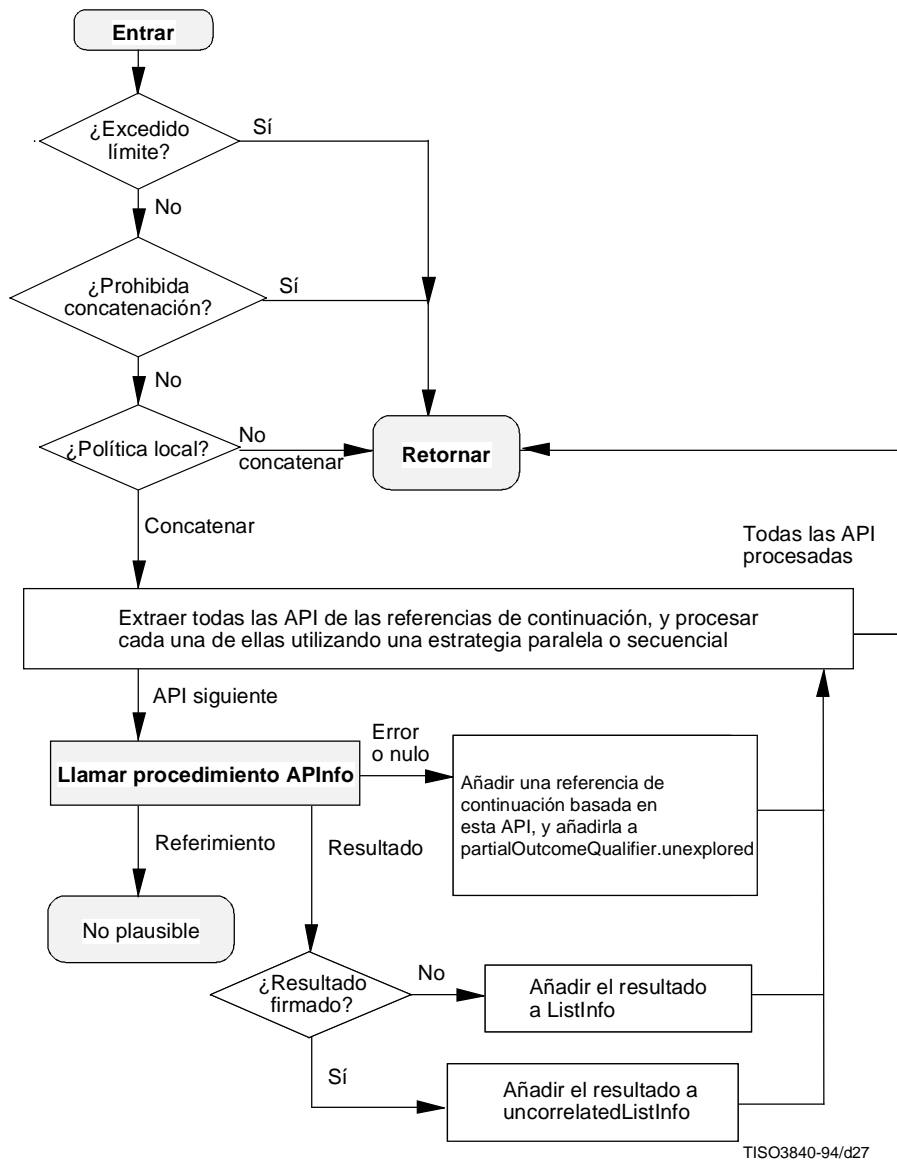


Figura 25 – Procedimiento List Continuation Reference (listar referencia de continuación)

Se ejecutarán los siguientes pasos:

- 1) Si hasta el momento se ha excedido cualquiera de los límites (problemas), retornar al **despachador de operaciones** para continuar con el procedimiento  **fusión de resultados**.
- 2) Si la bandera  **chainingProhibited** en  **CommonArguments.serviceControls** está fijada o el DSA decide no realizar concatenación alguna en razón de su política operacional local, el DSA retornará directamente al **despachador de operaciones** para proseguir con el procedimiento  **fusión de resultados**.
- 3) Crear un conjunto de valores de  **AccessPointInformation** para el componente  **accessPoint** de cada una de las referencias de continuación de la  **SRcontinuationList**.

Utilizar la estrategia paralelo o secuencial para procesar cada **API** del siguiente modo:

- i) Llamar el procedimiento **APIInfo** con la siguiente **API** en el conjunto.
- ii) Si se retorna un resultado, añadirlo a **listInfo** si no está firmado, o añadirlo a **uncorrelatedListInfo** si está firmado.
- iii) Si el retorno es un error o nulo, significa que **APIInfo** ya ha tratado con todos los puntos de acceso del **API**, sin éxito. Por razones de política local operacional o de seguridad, o bien ignorarlo y pasar a la siguiente **API**, o bien añadir una referencia de continuación sobre esta **API** al **partialOutcomeQualifier**.

NOTA – No es plausible obtener un referimiento de vuelta desde **APIInfo**. Cualquier "referimiento" debe venir en la forma de **unexplored** en **partialOutcomeQualifier**.

- 4) Una vez procesadas todas las **API**, retornar al **despachador de operaciones**.

### 20.4.3 Procedimiento Search Continuation Reference (buscar referencia de continuación)

El procedimiento **buscar referencia de continuación** consta de los pasos mostrados en la figura 26. Este procedimiento se invoca cuando no puede satisfacerse una petición de buscar en el DSA local y se ha añadido un conjunto de referencias de continuación a **SRcontinuationList** para concatenación o referimiento. El procedimiento es muy similar al procedimiento **listar referencia de continuación**. La diferencia es que en este caso las referencias de continuación de **SRcontinuationList** pueden tener diferentes valores **targetObject**. De este modo las referencias de continuación son clasificadas en conjuntos de referencias de continuación con el mismo **targetObject**. Asimismo, se define el uso de **exclusiones** en argumentos de concatenación y de **alreadySearched** en resultados de concatenación, pues ésta es una estrategia importante para la búsqueda. El uso de **exclusions** y **alreadySearched** se aplica al procesamiento de cada conjunto de referencia de continuación con el mismo **targetObject**.

Se ejecutarán los siguientes pasos:

- 1) Si se ha excedido hasta el momento cualquiera de los límites (problemas), retornar al **despachador de operaciones** para continuar con el procedimiento **fusión de resultados**.
- 2) Si la bandera **chainingProhibited** en **CommonArguments.serviceControls** está fijada o el DSA decide no realizar concatenación en razón de su política operacional local, el DSA retornará directamente al **despachador de operaciones** para continuar con el procedimiento **fusión de resultados**.
- 3) Clasificar las referencias de continuación de **SRcontinuationList** en conjuntos con el mismo **targetObject**. Dentro de cada conjunto, elimínese cualquier duplicado.

NOTA 1 – Si uno o más valores de **targetObject** no es un RDN primario, esta clasificación podría no ser precisa. En la clasificación se tendrán en cuenta los RDN distinguidos alternativos, en caso de que se conozcan.

- 4) Para cada subconjunto de referencias de continuación, crear un conjunto de valores de **AccessPointInformation** tomados del componente **accessPoints** de cada referencia de continuación del subconjunto, y elegir una estrategia secuencial o paralelo para continuar procesando. Si se elige la estrategia paralelo, sáltense los pasos que vienen a continuación, que sólo son aplicables a la estrategia secuencial.
  - a) Si se elige la estrategia secuencial, mantener una variable local **localExclusions** para cada conjunto de referencias de continuación con el mismo **targetObject**. Inicialmente **localExclusions** se fija a las **exclusions** de la petición de concatenación entrante (si existe) y se efectúa la búsqueda en todos los subárboles directamente bajo **targetObject**.
  - b) Si se utiliza la estrategia secuencial, comparar el **targetObject** con todos los elementos de **localExclusions**, y eliminar aquellos elementos que no contengan como prefijo **targetObject**. Estas son exclusiones relevantes para el actual objeto de destino.
  - c) Extraer todas las **API** de las referencias de continuación conjunto del actual de objeto de destino.
  - d) Formar un bucle que recorra todas las **API**. Para cada **API**:
    - i) Llamar **APIInfo**.
    - ii) Si se retorna un resultado, añadir el resultado a **searchInfo** si no está firmado, o añadirlo a **uncorrelatedSearchInfo** si está firmado. Si se utiliza la estrategia secuencial, actualizar **localExclusions** utilizando **alreadySearched** en la respuesta entrante, y también fusionar el **alreadySearched** de la respuesta entrante con el **ChainingResults.alreadySearched** de este DSA. Pasar luego a la siguiente **API**.

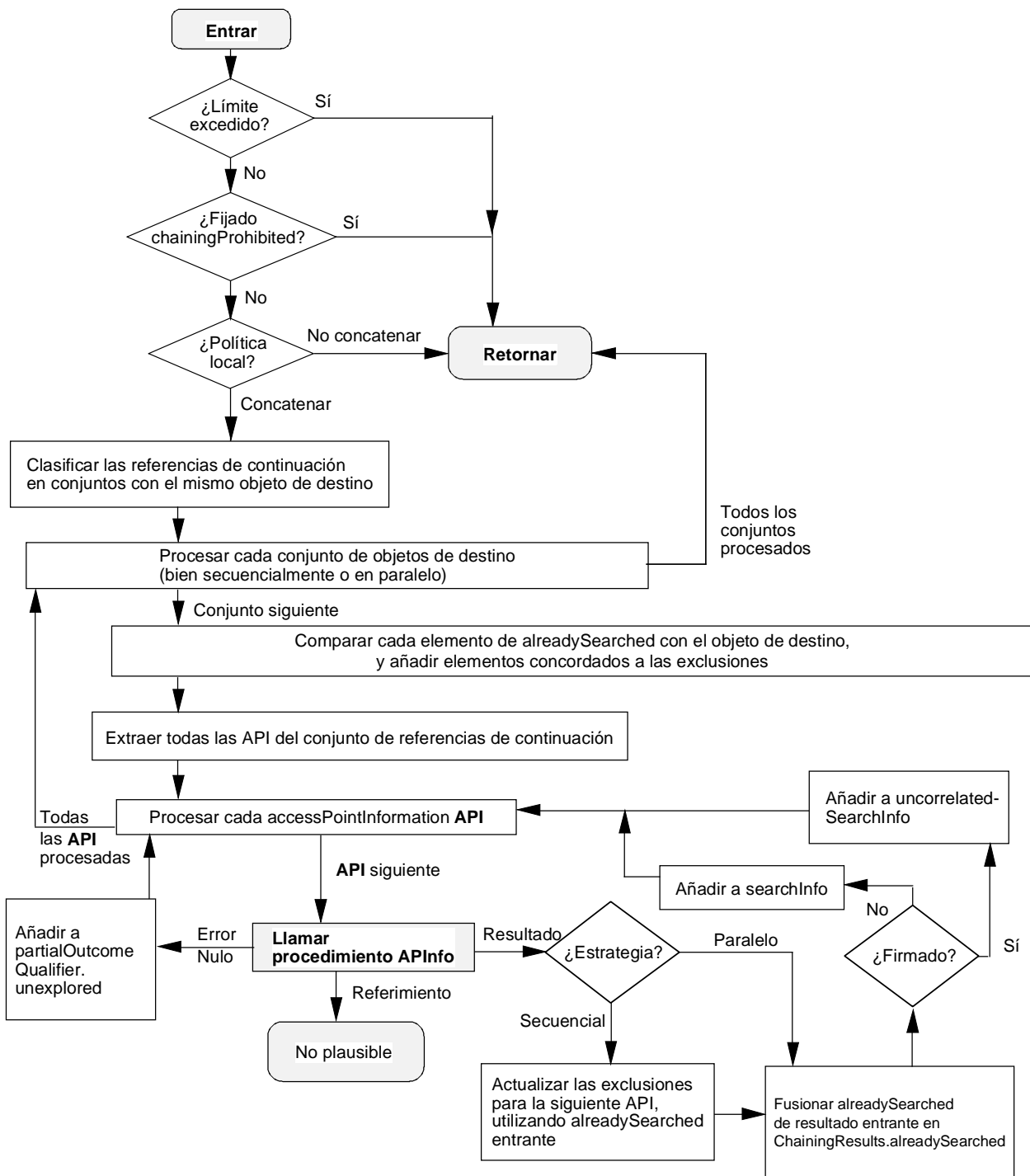


iii) Si se retorna un error o nulo, ello significa que **APIInfo** ya ha probado con todos los puntos de acceso del **API**, sin éxito. Por razones de política local operacional o de seguridad, o bien ignorarlo y pasar a la siguiente **API**, o añadir una referencia de continuación basada en esta **API** al **partialOutcomeQualifier**.

NOTA 2 – No es plausible obtener un referimiento de vuelta desde **APIInfo**. Cualquier "referimiento" debe venir en la forma de **unexplored** en **partialOutcomeQualifier**.

e) Una vez procesadas todas las **API**, pasar al siguiente conjunto de referencias de continuación con el mismo **targetObject**.

5) Una vez procesadas todas las referencias de continuación, retornar al **despachador de operaciones**.



TISO3850-94/d28

Figura 26 – Procedimiento Search Continuation Reference (buscar referencia de continuación)

#### 20.4.4 Procedimiento APInfo

Se llama este procedimiento para procesar una **AccessPointInformation**, que contiene uno o más puntos de acceso (véase la figura 27). Estos se procesan uno a uno hasta que se retorna un resultado o un error. Si el error es un error de servicio tal que puede tener éxito probar con otro punto de acceso, se prueba con puntos de acceso adicionales siempre que lo permitan la política operacional local.

- 1) Efectuar la detección de bucle. Si se detecta un bucle, retornar **serviceError** con el problema **loopDetected**. En caso contrario, continuar en el paso 2).
- 2) Procesar cada uno de los puntos de acceso según la información de punto de acceso. Si se han procesado todos, retornar un **null result**. Si hay algún punto de acceso por procesar, continuar en el paso 3).
- 3) Comprobar si la política local permite la concatenación a este punto de acceso. Esta comprobación debe tener en cuenta los valores fijados para los controles de servicio y los argumentos de concatenación (por ejemplo **chainingProhibited**, **preferChaining**, si el AP está o no dentro del **localScope**, **excludeShadows**, etc.). Si la política local o los valores fijados a los controles de servicio respectivos no permiten el uso de este punto de acceso particular, ignorar el punto de acceso y continuar en el paso 2). Si puede utilizarse el punto de acceso, continuar en el paso 4).
- 4) Si por la política local se seleccionó la estrategia sólo maestro, el argumento **excludeShadows** se fija a **TRUE**.

Si **nameResolutionsPhase** no está **completed** y la estrategia prescribe que se continúe la resolución de nombre sobre inserciones maestros, se fija **nameResolveOnMaster** a **TRUE**.

El argumento de concatenación **nameResolveOnMaster** se fijará a **TRUE** si se da cualquiera de los casos siguientes:

- en el argumento de concatenación entrante **nameResolutionPhase** es **proceeding** y **nameResolveOnMaster** es **TRUE**; o
- la operación es una de las operaciones de modificación, el **referenceType** de la petición de concatenación que ha de emitirse es NSSR, y se utiliza una estrategia paralelo.

NOTA – Este método basado en **nameResolveOnMaster** se utiliza para impedir que operaciones de modificación se apliquen en múltiples ocasiones debido a la presencia de NSSR.

- 5) Construir una petición concatenada e intentar emitirla:
  - a) Realizar evitación de bucle comprobando si un ítem con el mismo **targetObject** y **operationProgress** aparece en **tracelInformation** del **ChainingArguments** recibido. Si la petición resultante [según se describe en el paso 5) c) tendría por resultado un bucle, el DSA retornará o bien un **serviceError** con el problema **loopDetected** al DUA/DSA solicitante, o ignorará el punto de acceso, probará con el siguiente, y continuará en el paso 2].
  - b) Si la petición o subpetición que se ha de concatenar es el resultado de la ejecución de un referimiento, se requiere una verificación suplementaria para evitación de bucle. Verificar si un ítem con el mismo **targetObject**, **operationProgress** y DSA de destino tiene lugar en **referralRequests**. Si así fuera, seguir las indicaciones del apartado a). Si así no fuera, añádase un nuevo **Traceltem** a **referralRequests** con los siguientes componentes:
    - **targetObject** y **operationProgress** fijados en el valor de la petición/subpetición concatenada;
    - **dsa** fijado al nombre del DSA al que la petición/subpetición ha de ser concatenado.
  - c) Tras un Bind exitoso, el DSA realizará una operación concatenada del mismo tipo de operación que la procesada con los siguientes parámetros:
    - el argumento de la operación concatenada se fija como el argumento de operación recibido,
    - **ChainingArguments.originator** se fija como se ha recibido,
    - **ChainingArguments.targetObject** se fija al **targetObject** de la referencia de continuación,
    - **ChainingArguments.operationProgress** se fija al valor de **operationProgress** de la referencia de continuación,
    - **ChainingArguments.tracelInformation** se fija a información de rastreo actualizada por el procedimiento **solicitar validación**,
    - **ChainingArguments.aliasDereferenced** se fija al valor actualizado del **aliasDereferenced** actualizado localmente,
    - **ChainingArguments.returnCrossRefs** se fija a una opción local,

- **ChainingArguments.referenceType** se fija al valor de **referenceType** de la referencia de continuación,
  - **ChainingArguments.timeLimit** se fija al valor del **timeLimit** recibido,
  - **ChainingArguments.exclusions**, ausente,
  - **SecurityParameters** se fija al valor de los **SecurityParameters** recibidos.
- 6) Si no pudo emitirse con éxito la petición, continuar en el paso 7). De lo contrario, continuar en el paso 8).
- 7) Es una opción local la de si se ha de continuar o no. Si el DSA opta por continuar, se ignora el error y se intenta el siguiente punto de acceso. Seguir en el paso 2). Si el DSA decide no intentar otro punto de acceso, es una cuestión de política local la de retornar o no un correspondiente **referral** o un **serviceError** al llamante del procedimiento.
- 8) Si pudo emitirse con éxito la petición, el DSA esperará la respuesta y la procesará:
- a) si se recibe un **result**, se retorna el **result** al llamante del procedimiento,
  - b) si se recibe un **serviceError** con el problema **busy**, **unavailable**, **unwillingToPerform** o **invalidReference**, continuar en el paso 7),
  - c) si se recibe **referral** y **returnToDUA** está fijado a **TRUE**, el DSA receptor no actuará sobre el referimiento, sino que lo devolverá al peticionario,
  - d) si se recibe un **referral** y **returnToDUA** está fijado a **FALSE**, son aplicables las mismas consideraciones de política local que en el paso 3) (teniendo en cuenta los controles de servicio, argumentos de concatenación, estrategias de concatenación, etc.). Si se decide no desreferenciar el **referimiento**, se retorna el **referimiento** al llamante. Si se decide desreferenciar el **referimiento**, se vacía la **NRcontinuationList**, se coloca la referencia de continuación en **NRcontinuationList** tal como fue recibida en el **referimiento**, y se llama el procedimiento de **referencia de continuación de resolución de nombre**. Esto puede producir un **result**, **referral**, **serviceError** u otro error. Cualquiera que sea lo que se reciba como consecuencia de la llamada del procedimiento **referencia de continuación de resolución de nombre**, será devuelto al llamante,
  - e) si se produce cualquier otro **error**, deberá devolverse al llamante.

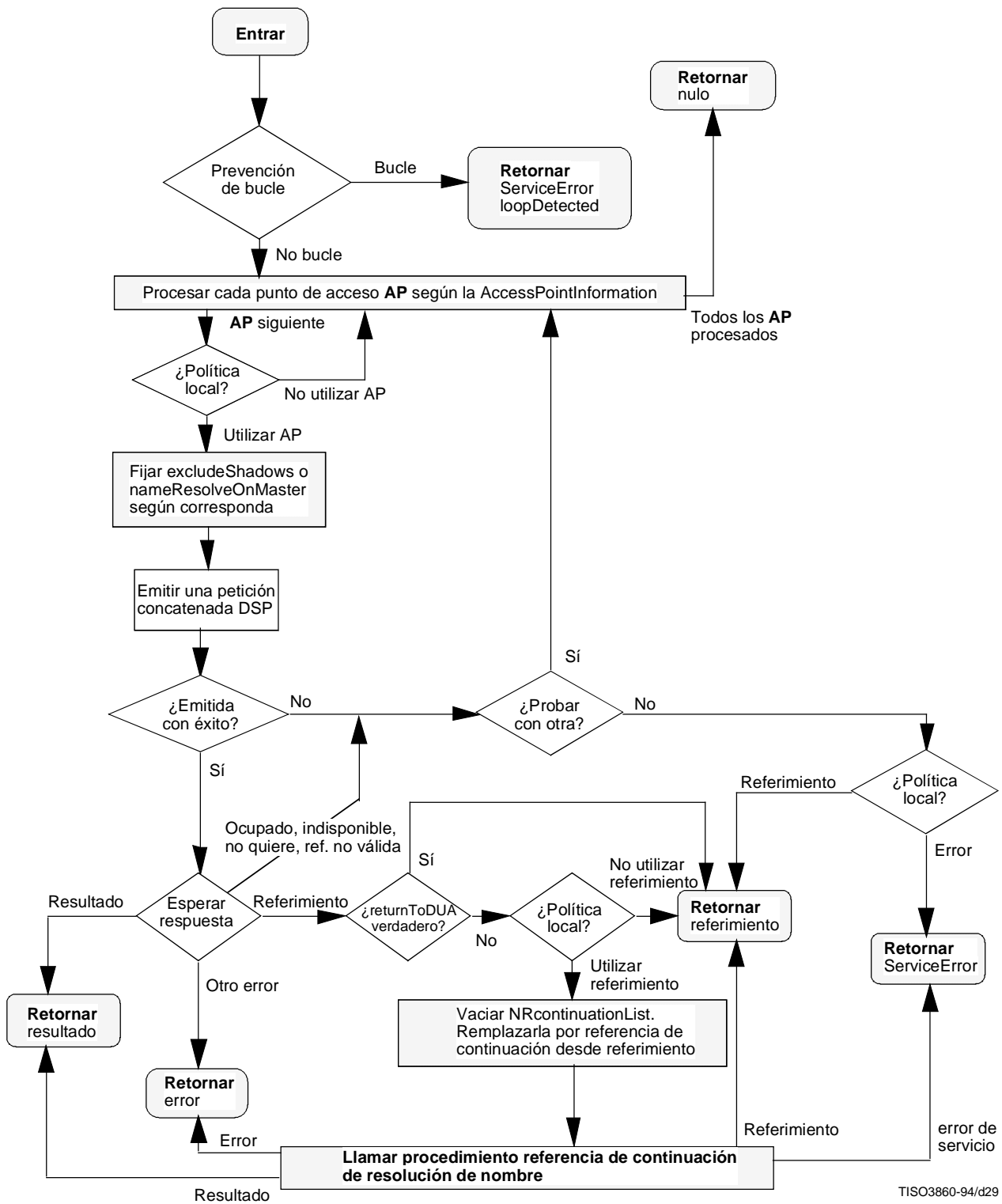
## 20.5 Procedimiento Abandon (abandonar)

Este procedimiento se invoca si se recibe una petición de **abandonar**. Consta de los siguientes pasos, según se muestra en la figura 28:

- 1) Cuando se recibe una petición de **abandonar**, que referencia una operación desconocida, se retornará al solicitante un **abandonFailed** con un valor de error **noSuchOperation**.
- 2) Si la petición que va a ser abandonada ya ha tenido una respuesta, y el DSA ha conservado la información por la cual conoce esta circunstancia, puede retornarse al solicitante un **abandonError** con un valor de error **tooLate**.
- 3) Si la petición de **abandonar** no es válida, es decir pide abandonar una petición que no es una petición de interrogación, se retornará al solicitante un **abandonFailed** con un valor de error **cannotAbandon**.
- 4) Si un DSA tiene (sub)peticiones concatenadas pendientes cuando recibe una petición válida de abandonar la petición original, y el DSA decide intentar abandonar, enviará peticiones de abandonar para ninguna, algunas o todas las (sub)peticiones pendientes para la operación en cuestión, y luego esperará la respuesta a la petición de abandonar y las (sub)peticiones pendientes. En cualquier momento del curso de esta operación, el DSA puede enviar un resultado de abandonar y un **abandonFailed** al solicitante y luego descartar las respuestas a las peticiones de abandonar emitidas y a (sub)peticiones pendientes a medida que llegan.

Si el DSA decide no enviar respuestas al solicitante hasta que no haya más (sub)peticiones pendientes, opcionalmente puede enviar un error **abandonedFailed** al solicitante si a todas las peticiones de abandonar emitidos se respondió con errores **abandonedFailed** y si no se ha realizado una operación local de abandono.

Si se retorna un **abandonedFailed** al solicitante, se tratará la petición original como si nunca se hubiese recibido la petición de abandonar.



TISO3860-94/d29

Figura 27 – Procedimiento APInfo

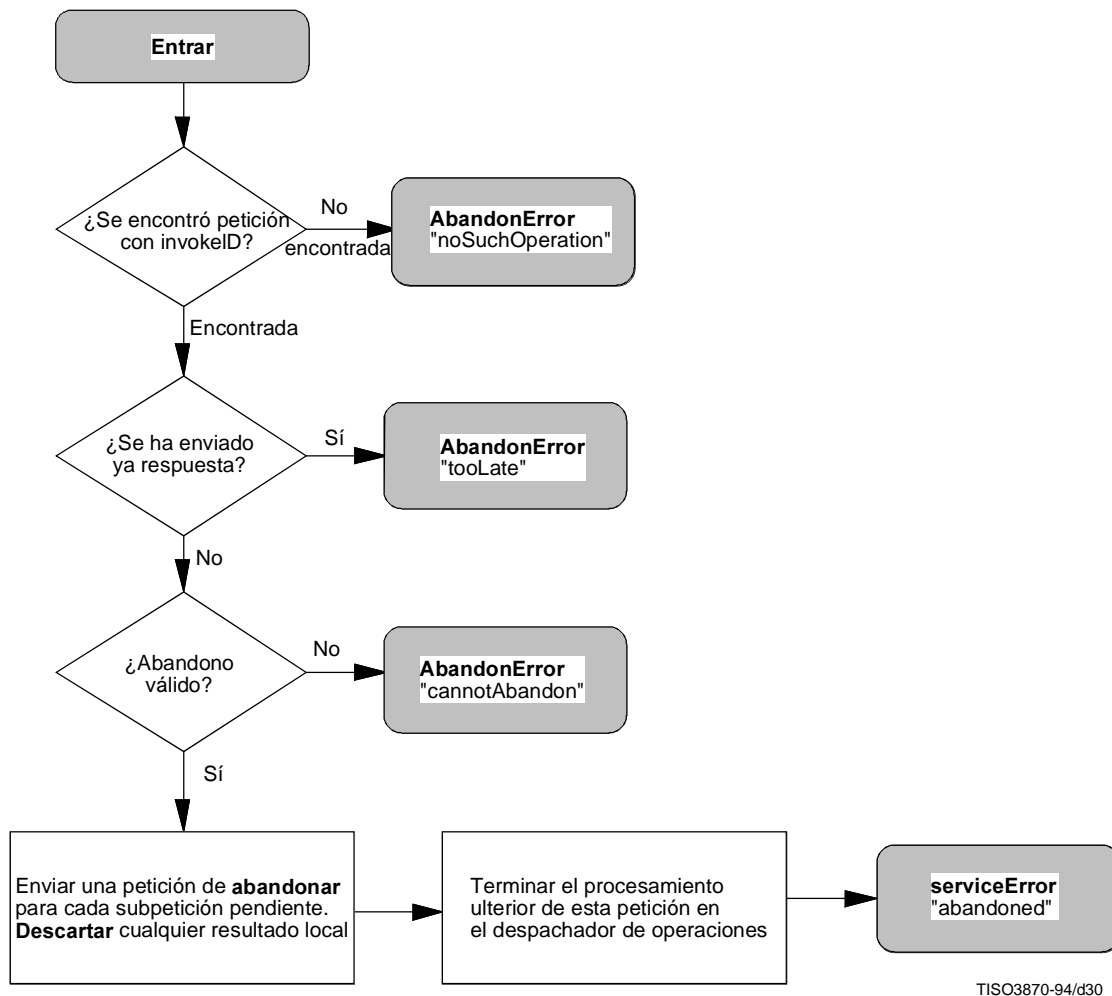


Figura 28 – Procedimiento Abandon (abandonar)

## 21 Procedimiento Results Merging (fusión de resultados)

El procedimiento **fusión de resultados** de la figura 29 se llama siguiendo uno de los procedimientos de **referencia de continuación**. Si el resultado no está firmado, este procedimiento elimina los duplicados; y si existen referencias de continuación adicionales en **partialOutcomeQualifier.unexplored** se llama el procedimiento o los procedimientos de referencia de continuación relevantes si lo permite la política operacional local:

- 1) Si la operación es una operación de listar, continuar en el paso 2); si la operación es una operación de buscar, continuar en el paso 3); en otro caso retornar el resultado que se suministró como parámetro de entrada al procedimiento **fusión de resultado**.
- 2) La operación es una operación de listar. Suprimir todos los duplicados, dando preferencia a la información maestra sobre la información sombreada.

Si el resultado de la operación se generó localmente y contiene referencias de continuación, éstas no se utilizarán para concatenación sino que se retornarán al usuario. En este caso, continuar en el paso 6).

Si el resultado de la operación se recibió como resultado de una operación de listar concatenada, el resultado puede contener referencias de continuación. En este caso, comprobar si el usuario había fijado el control de servicio **preferChaining**. Si es **TRUE**, las referencias de continuación deberán ser utilizadas para concatenación por el DSA. Continuar en el paso 4).

- 3) La operación es una operación de buscar. Suprimir todos los duplicados, dando preferencia a la información maestra sobre la información sombreada. Si hay un problema de límite, se retorna el resultado. En otro caso se continúa en el paso 4).

- 4) Procesar cada una de las referencias de continuación que se encuentra en el **partialOutcomeQualifier.unexplored** del resultado de toda operación concatenada. Si la política local prescribe no utilizarla para concatenación, se ignorará y se elegirá otra referencia de continuación. Si la política local autoriza la utilización de la referencia de continuación para concatenación, se efectúa lo siguiente:

Se comprueba la **nameResolutionPhase** que se suministró en la referencia de continuación. Si es **notStarted** o **proceeding**, se añade a la lista de referencias de continuación que será suministrada al procedimiento de **continuación de resolución de nombre (NRcontinuationList)**. Si **nameResolutionPhase** está **completed**, se añade la referencia de continuación a la lista de referencias de continuación que se ha suministrado al procedimiento de continuación de la petición (**SRcontinuationList**).

Se prosigue hasta que hayan sido procesadas todas las referencias de continuación.

- 5) Si no hay referencias para procesar en **SRcontinuationList**, se comprueba el tipo de operación. Si la operación es una operación de listar, se llama el procedimiento de **listar referencia de continuación** y se continúa en el paso 2). Si la operación es una operación de buscar, se llama el procedimiento de **buscar referencia de continuación** y se continúa en el paso 3).

Si **SRcontinuationList** está vacía, se comprueba si hay referencias de continuación en **NRcontinuationList**. Si las hay, se llama el procedimiento de **referencia de continuación de resolución de nombre** y se continúa en el paso 3).

Si ambas listas de continuación están vacías, se continúa en el paso 6).

- 6) Se comprueba si el resultado está vacío. Si no lo está, se devuelve dicho resultado. Si está vacío, o bien se retorna un resultado nulo si el control de acceso y la política local lo permiten, o se retorna un error apropiado.

Si se efectúa la protección, no se efectuará la fusión de los resultados.

## 22 Procedimientos para autenticación distribuida

En esta cláusula se especifican los procedimientos necesarios para prestar los servicios de autenticación distribuida de directorio. Estos servicios, y por tanto los procedimientos, se dividen en las dos categorías siguientes:

- autenticación de originador, efectuada de forma no protegida (basada en la simple identidad) o securizada (basada en firmas digitales); y
- autenticación de resultados, protegidos de forma similar (basada también en firmas digitales).

### 22.1 Autenticación de originador

#### 22.1.1 Autenticación basada en la identidad

El servicio de autenticación basado en la identidad permite a los DSA autenticar al solicitante original de información con el fin de efectuar controles de acceso local. Los DSA que deseen explotar este servicio deben utilizar el siguiente procedimiento:

- Un DSA que deba autenticar una petición DAP obtiene el nombre distinguido del solicitante mediante los procedimientos Bind (vincular) en el momento en que se establece una asociación del DUA (con el DSA). El éxito en la conclusión de estos procedimientos no prejuzga, en manera alguna, el nivel de autenticación que puede exigirse ulteriormente para procesar operaciones utilizando esa asociación.
- El DSA con el que está asociado el DUA debe insertar el nombre distinguido del solicitante en el campo iniciador del **ChainingArguments** en todas las operaciones concatenadas ulteriores para otros DSA.
- Cuando un DSA reciba una operación concatenada podrá ejecutarla 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 problema **insufficientAccessRights**.

#### 22.1.2 Autenticación de originador basada en la firma

El servicio de autenticación de originador basada en la firma permite a un DSA autenticar (de una manera securizada) al originador de una determinada petición de servicio. En esta cláusula se describen los procedimientos que sigue un DSA para proporcionar este servicio.

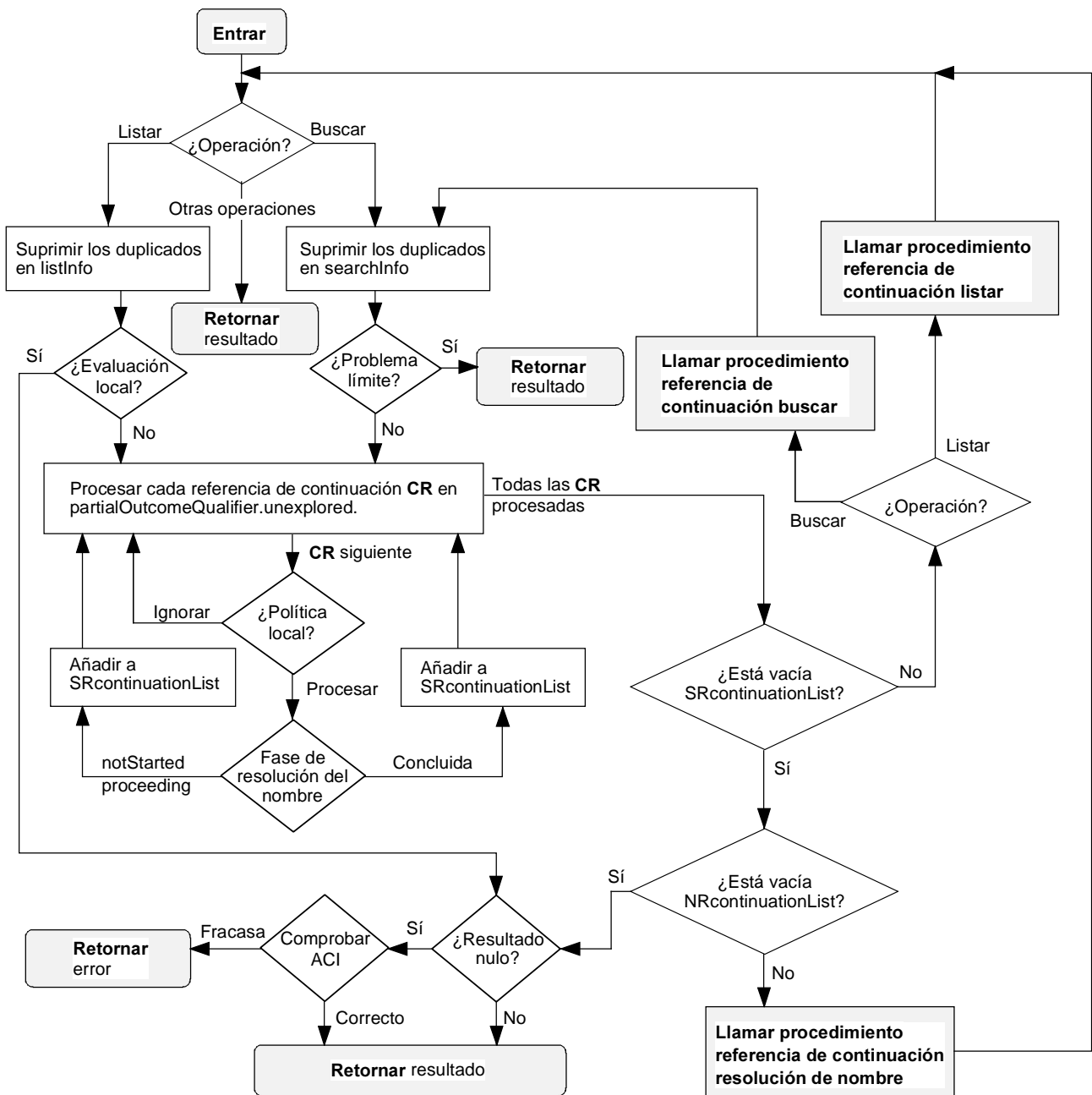
El servicio de autenticación basado en la firma es invocado por un DUA utilizando la variante **PROTECTED** de una petición de servicio protegida facultativamente con **DirQOP signed** o **signedAndEncrypted**.

Un DSA que reciba una petición firmada procedente de otro DSA, suprimirá la firma de este último antes de procesar la operación. Suponiendo que el resultado de cualquier verificación de la firma es satisfactorio, el DSA proseguirá la operación. Si, durante el procesamiento, el DSA necesita realizar una concatenación, el conjunto de argumentos para cada operación concatenada se construirá de la siguiente manera:

- El DSA forma un conjunto de argumentos que pueden ser firmados facultativamente, constituido por el conjunto de argumentos firmados entrante junto con un **ChainingArguments** modificado.

Si el DSA es capaz de aportar información a la respuesta, puede emplearse autenticación de originador, basada en la petición de servicio firmada, para determinar los derechos de acceso a esa información.

Si un DSA recibe una petición de servicio no firmada de información que sólo se da a conocer tras la autenticación del originador, se retorna a un **securityError** con **SecurityProblem** fijado a **protectionRequired**.



TISO3880-94/d31

Figura 29 – Procedimiento Results Merging (fusión de los resultados)

## 22.2 Autenticación de resultados

Ese servicio se proporciona para que los solicitantes de operaciones de directorio (sean estos DUA o DSA) puedan verificar (de una manera securizada utilizando técnicas de firma digital) la fuente 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 firmado del componente **protectionRequest** tal como figura en el conjunto de argumentos de operaciones de directorio; una DSA que reciba una operación, en la que se haya seleccionado esta opción, puede firmar, facultativamente, cualesquiera resultados ulteriores. La opción firmada en la petición de protección sirve de indicación a el DSA de la preferencia del peticionario; de hecho, el DSA puede firmar o no cualesquiera resultados ulteriores.

Cuando un DSA realiza una concatenación tiene varias opciones 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 DSA, o aportadas por este DSA, o de ambas procedencias.

## SECCIÓN 6 – ADMINISTRACIÓN DE CONOCIMIENTO

### 23 Visión de conjunto de la administración de conocimiento

Para operar un directorio ampliamente distribuido con un grado de consistencia y efectividad aceptables se necesitan procedimientos para crear, mantener y extender el conocimiento que tiene cada DSA. Se utilizan los siguientes mecanismos conjuntamente para administrar el conocimiento de un DSA.

- a) *Vinculaciones operacionales jerárquicas y jerárquicas no específicas* – Estos procedimientos y protocolos se definen en las cláusulas 24 y 25. Se utilizan para crear y mantener referencias subordinadas, referencias subordinadas no específicas y referencias superiores inmediatas, así como la información de prefijo de contexto para contextos de denominación. Estas vinculaciones operacionales se establecen entre DSA maestros que tienen contextos de denominación jerárquicamente relacionados entre sí como subordinado inmediato a superior inmediato. Los procedimientos pueden ser puestos en marcha como un efecto secundario de la modificación del RDN de una inserción o de la adición o supresión de una inserción, cuyo superior inmediato no está contenido en el mismo DSA que contiene la inserción en cuestión.
- b) *Vinculaciones operacionales de sombreado* – Estos procedimientos y protocolos se definen en la Rec. UIT-T X.525 | ISO/CEI 9594-9. Se utilizan para crear y mantener referencias de conocimiento de dos maneras. En la primera, que viene a ser un efecto secundario de establecer (o terminar) acuerdos de sombreado, puntos de acceso se añaden (o suprimen) en el **consumerKnowledge** y opcionalmente en los atributos operacionales **secondaryShadow**. Esta información puede ser entonces utilizada por los procedimientos y protocolos discutidos anteriormente para actualizar la referencia subordinada en el DSA maestro superior y la referencia superior inmediata en el DSA maestro subordinado. En la segunda, el DISP propaga las referencias de conocimiento contenidas por DSA maestros a DSA consumidores de sombra.
- c) *Referencias de remisión* – La distribución de referencias de remisión es una prestación del DSP. Su utilización para crear y mantener referencias de remisiones se recapitula en 23.2.

NOTA – Los mecanismos para inicializar y mantener la referencia superior y **myAccessPoint** quedan fuera del ámbito de esta Especificación de directorio.

#### 23.1 Mantenimiento de referencias de conocimiento

Esta subcláusula describe cómo se utiliza el DOP para mantener atributos operacionales del DSA que expresan conocimiento. En el anexo E se presenta un ejemplo simple de la relación entre atributos de conocimiento y los protocolos empleados para mantenerlos.



### 23.1.1 Mantenimiento de conocimiento de consumidor por DSA suministradores y maestros

Una referencia de consumidor se expresa mediante un valor del atributo **consumerKnowledge**, contenido por un DSA suministrador de sombra y asociado con el prefijo de contexto para un contexto de denominación; una referencia de suministrador, mediante un valor del atributo **supplierKnowledge**, contenido por un DSA consumidor de sombra y también asociado con el contexto de prefijo para un contexto de denominación. Ambos atributos están contenidos en las DSE de tipo **cp**. Al establecerse la vinculación operacional de sombra se crea un valor de cada uno de estos atributos, los cuales son actualizados al modificar la vinculación operacional de sombra.

Un DSA suministrador puede obtener la información para construir valores del atributo **secondaryShadows** si el componente **secondaryShadows** opcional de su **ShadowingAgreementInfo** con un consumidor es **TRUE**. En este caso, cuando el DSA consumidor detecta que el conjunto de DSA que contiene copias del área replicada utilizable en común (sus consumidores, o, a su vez, los consumidores de sus consumidores, etc., cualquiera que sea la profundidad del sombreado secundario) ha cambiado (por adición, modificación o supresión de puntos de acceso), comunica esta nueva información (un conjunto de **SuppliersAndConsumers**) por medio de una operación **modifyOperationalBinding**, como se describe en la Rec. UIT-T X.525 | ISO/CEI 9594-9.

Un DSA suministrador mantiene su propio atributo **secondaryShadows** asociado con el prefijo de contexto de la manera siguiente:

- a) El conjunto de **SuppliersAndConsumers** recibido de un consumidor por medio de una operación **modifyOperationalBinding** puede utilizarse para crear, o reemplazar valores del atributo. El componente suministrado de **SuppliersAndConsumers** representa el punto de acceso de un DSA consumidor (o de sus consumidores, etc., lo que depende de la profundidad del sombreado secundario); el componente consumidores representa el conjunto de consumidores del consumidor (o de sus consumidores etc., según la profundidad del sombreado secundario).
- b) Todo consumidor que proporciona a su suministrador una operación **modifyOperationalBinding** que contiene un conjunto de **SuppliersAndConsumers**, incluye los siguientes valores: los valores de su atributo **secondaryShadows**, y un valor de nueva creación. Este valor se construye utilizando su propio punto de acceso, **myAccessPoint** (como el componente suministrador), y los valores de los puntos de acceso de los consumidores, contenidos en el atributo **consumerKnowledge**, que representan consumidores que contienen sombras utilizables en común (como el componente consumidores).

Una utilización recursiva de este procedimiento permite a un DSA maestro para un contexto de denominación tener conocimiento de todos sus DSA consumidores de sombra secundaria que contienen áreas replicadas utilizables en común derivadas del contexto de denominación. Esta información está entonces disponible para el mantenimiento de referencias subordinadas, referencias subordinadas no específicas, y referencias de superior inmediato.

### 23.1.2 Mantenimiento de conocimiento subordinado y superior inmediato en DSA maestros

Una referencia subordinada se expresa mediante un valor del atributo **specificKnowledge**, contenido en una DSE de tipo **subr** por el DSA que contiene el contexto de denominación inmediatamente superior al referenciado; una referencia superior inmediata, mediante un valor del atributo **specificKnowledge**, contenido en una DSE de tipo **immSupr** por el DSA que contiene el contexto de denominación inmediatamente subordinado al referenciado. Un valor de cada uno de estos atributos es creado en los DSA maestros superior y subordinado al establecerse la HOB, y actualizado al modificar la HOB.

Un DSA maestro subordinado proporciona a un DSA maestro superior la información para construir su referencia subordinada vía el componente **accessPoints** del parámetro **subordinateToSuperior** que transfiere al superior en el DOP. La información incluida en **accessPoints** se determina por valores de atributos contenidos por el DSA subordinado, de la forma siguiente:

- a) El valor del atributo **myAccessPoint** (contenido en la DSE raíz) se utiliza para formar el elemento en **accessPoints** con **category** de valor **master**.
- b) Los valores del **consumerKnowledge** y **secondaryShadows** (contenidos ambos en la DSE de prefijo de contexto subordinado) se utilizan para formar elementos adicionales en **accessPoints** con **category** de valor **shadow**.

Un DSA maestro superior proporciona a un DSA maestro subordinado la información para construir su referencia subordinada vía el componente **contextPrefixInfo** del parámetro **SuperiorToSubordinate** que transfiere al subordinado en el DOP. Este componente es un valor de tipo **SEQUENCE OF Vertex**, que contiene una secuencia de elementos que corresponde al trayecto desde la raíz del DIT hasta el prefijo de contexto subordinado. Para uno de estos elementos, que

corresponde al prefijo de contexto del contexto de denominación del superior inmediato, el componente opcional **accessPoints** estará presente. El DSA subordinado contiene esta información como un atributo **specificKnowledge** en la DSE, de tipo **immSupr**, que corresponde a este elemento de **contextPrefixInfo**. La información incluida en **accessPoints** por el DSA superior se determina por valores de atributos contenidos en el DSA superior de la forma siguiente:

- a) El valor de atributo **myAccessPoint** (contenido en la DSE raíz) se utiliza para formar el elemento en **accessPoints** con **category** de valor **master**.
- b) Los valores de los atributos **consumerKnowledge** y **secondaryShadows** (contenidos ambos en la DSE de prefijo de contexto superior) se utilizan para formar elementos adicionales en **accessPoints** con **category** de valor **shadow**.

NOTA – Solamente aquellos puntos de acceso que correspondan a DSA consumidores que reciben áreas replicadas utilizables en común deben ser seleccionados por los DSA superior y subordinado, a partir de sus atributos **consumerKnowledge** para inclusión en **accessPoints**. Los procedimientos para la construcción de **secondaryShadows** garantizan que estos puntos de acceso identificarán DSA de sombra que contienen áreas replicadas utilizables en común.

### 23.1.3 Mantenimiento de conocimiento subordinado e inmediatamente superior en DSA consumidores

Un DSA consumidor de sombra que está contratando con su suministrador para recibir los conocimientos superior y subordinado inmediatos, asociados con una unidad de replicación, contrata efectivamente mediante el DISP, de modo que sus referencias superior y subordinada inmediatas sean mantenidas por su DSA suministrador de sombra.

NOTA – Para ciertas especificaciones de unidad de replicación, puede ser necesario que el DSA consumidor contrate para recibir **extendedKnowledge** a fin de que su suministrador pueda proporcionarle conocimiento subordinado.

## 23.2 Petición de referencia de remisión

Para mejorar la efectividad del sistema de directorio, el conjunto local de referencias de remisión puede ampliarse mediante operaciones ordinarias de directorio. Si un DSA soporta el DSP, puede pedir a otro DSA (que también deberá soportar el DSP) que retorne aquellas referencias de conocimiento que contienen información sobre la ubicación de los contextos de denominación relacionados con el nombre de objeto de destino de una operación ordinaria de directorio.

Si el componente **returnCrossRefs** del **ChainingArguments** está fijado a **TRUE**, puede estar presente el componente **crossReference** del **ChainingResults**, compuesto de una secuencia de ítems de referencias de remisión.

Si un DSA no es capaz de concatenar una petición al siguiente DSA, se retorna un referimiento al DSA originador. Si el componente **returnCrossRefs** de **ChainingArguments** era **TRUE**, el referimiento puede contener además el prefijo del contexto de denominación al que él refiere. El componente **contextPrefix** está ausente, si el referimiento se basa en una referencia subordinada no específica. La referencia de remisión retornada por un referimiento se basa en el conocimiento contenido por el DSA que generó el referimiento.

En ambos casos (resultado de concatenación y referimiento) una autoridad administrativa mediante su DSA puede optar por ignorar la petición de retornar referencias de remisión.

## 23.3 Inconsistencias de conocimiento

El directorio tiene que soportar mecanismos de comprobación de la consistencia para garantizar un cierto grado de consistencia del conocimiento.

NOTA – En ciertas circunstancias, una referencia de conocimiento será exacta (no válida en los sentidos antes mencionados) pero no válida para uso en un DSA porque el DMD (dominio de gestión de directorio) del DSA referenciado no desea ser contactado en lo absoluto por el DSA referenciante (por ejemplo, un DSA que ha adquirido, de cierta forma, una referencia de remisión al DSA referenciado) o no desea ser contactado en un rol particular (por ejemplo, como el DSA maestro para un contexto de denominación).

### 23.3.1 Detección de inconsistencias de conocimiento

El género de inconsistencia y su detección varía según los diferentes tipos de referencias de conocimiento:

- a) *Referencias de remisión y subordinadas* – Este tipo de referencia es no válido si el DSA referenciado no tiene un contexto de denominación o un área replicada derivada del contexto de denominación con el prefijo de contexto contenido en la referencia. Esta inconsistencia se detectará durante el proceso de resolución del nombre por inspección de los componentes **operationProgress** y **referenceType** de **ChainingArguments**.

- b) *Referencias subordinadas no específicas* – Este tipo de referencia es no válido si el DSA referenciado no tiene un contexto de denominación local con el prefijo de contexto contenido en la referencia menos el último RDN. La comprobación de consistencia se aplica como se ha dicho antes.
- c) *Referencias superiores* – Una referencia superior no válida es aquella que no forma parte del trayecto de referencia a la raíz. El mantenimiento de referencias superiores se realizará por medios externos y está fuera del ámbito de esta Especificación de directorio.

NOTA – No siempre es posible detectar una referencia superior no válida.

- d) *Referencias superiores inmediatas* – Este tipo de referencia es no válido si el DSA referenciado no contiene un contexto de denominación o un área replicada derivada del contexto de denominación con el prefijo de contexto contenido en la referencia. Además, la utilización de este tipo de referencia sólo es válida cuando el componente **operationProgress** de **ChainingArguments** tiene el valor **notStarted** o **proceeding**. Esta inconsistencia se detectará durante el proceso de resolución de nombre por inspección de los componentes **operationProgress** y **referenceType** de **ChainingArguments**.
- e) *Referencias de suministrador* – Este tipo de referencia, que identifica el suministrador de un área replicada y opcionalmente el maestro para el contexto de denominación del cual se derivó el área replicada, es no válido si el DSA referenciado no es el suministrador de sombra para el DSA que utiliza la referencia (cuando el componente **referenceType** de **ChainingArguments** tiene el valor **supplier**), o si el DSA referenciado no es el maestro para el contexto de denominación (cuando **referenceType** tiene el valor **master**). Esta inconsistencia se detectará durante las fases de resolución de nombre y evaluación de operación, en el procesamiento de la operación, por inspección del componente **referenceType** de **ChainingArguments**.

### 23.3.2 Señalización de inconsistencias de conocimiento

Si se utiliza la concatenación para realizar una petición de directorio, el DSA que tiene la referencia de conocimiento no válida detectará todas las inconsistencias de conocimiento al recibir un **serviceError** con el problema de **invalidReference**.

Si el DSA retorna un referimiento que se basó en una referencia de conocimiento no válida, el solicitante recibirá como retorno un **serviceError** con el problema de **invalidReference** si utiliza el referimiento. Está fuera del ámbito de esta Especificación de directorio la explicación de cómo se propagará la condición de error al DSA que almacena la referencia no válida.

### 23.3.3 Tratamiento de las referencias de conocimiento inconsistentes

Una vez que un DSA ha detectado una referencia no válida deberá intentar restablecer la consistencia de conocimiento. Por ejemplo, esto puede realizarse simplemente borrando una referencia recíproca no válida o sustituyéndola por una correcta que puede obtenerse utilizando los mecanismos **returnCrossRefs**.

La manera en que un DSA maneja realmente las referencias no válidas es un asunto local, y está fuera del ámbito de esta Especificación de directorio.

## 23.4 Referencias de conocimiento y contextos

Los nombres en las referencias de conocimiento serán nombres distinguidos primarios y pueden incluir valores distinguidos alternativos e información de contexto contenida en **valuesWithContext** para cualquier atributo que contribuya a cualquier RDN, según se describe en 9.3 de la Rec. UIT-T X.501 | ISO/CEI 9594-2.

Dependiendo del modo según el cual se obtiene una referencia de conocimiento (en particular si un DSA anterior a 1997 contiene la referencia o si éste ha formado parte de la cadena a través de la cual se ha obtenido la referencia), es posible que la referencia de conocimiento no incluya todos los posibles nombres distinguidos alternativos. Como resultado de ello, el nombre supuesto puede no ser reconocido como el mismo nombre por el titular de la referencia de conocimiento, lo que conduce a pasos adicionales en la resolución del nombre o, en ciertos casos, a resultados incoherentes o a la falla de la resolución del nombre. La utilización generalizada de los nombres distinguidos primarios, cuando se conocen, optimiza la capacidad de directorio para hacer frente a las variantes de contexto en los nombres.

## 24 Vinculaciones operacionales jerárquicas

Se utiliza una vinculación operacional jerárquica para representar la relación entre dos DSA que contienen dos contextos de denominación, uno inmediatamente subordinado al otro. En el caso de una HOB, el DSA superior contiene una referencia subordinada al contexto de denominación contenido por el DSA subordinado; el DSA subordinado contiene una referencia superior inmediata al contexto de denominación contenido por el DSA superior. La vinculación

operacional asegura que la información de conocimiento apropiada es intercambiada y mantenida entre los dos DSA de modo que ambos puedan comportarse durante el proceso de resolución de nombre y evaluación de operación como se define en las cláusulas 18 y 19.

## 24.1 Características de tipo vinculación operacional

### 24.1.1 Simetría y roles

El tipo vinculación operacional jerárquica es un tipo asimétrico de vinculación operacional. Los dos roles en la vinculación de este tipo son:

- a) el rol del DSA maestro para el contexto de denominación superior, el *DSA superior* (asociado con el rol abstracto "A"); y
- b) el rol del DSA maestro para el contexto de denominación subordinado, el *DSA subordinado* (asociado con el rol abstracto "B").

### 24.1.2 Acuerdo

La información de acuerdo intercambiada durante el establecimiento de la vinculación operacional jerárquica es un valor de **HierarchicalAgreement**. Contiene el nombre distinguido relativo del nuevo prefijo de contexto (el componente **rdn**) y el nombre distinguido de la inserción superior inmediata del nuevo contexto de denominación (el componente **immediateSuperior**). Esta información será proporcionada por el DSA que inicia la HOB.

```
HierarchicalAgreement ::= SEQUENCE {
    rdn                [0] RelativeDistinguishedName,
    immediateSuperior [1] DistinguishedName }
```

El **rdn** será el RDN primario, e **immediateSuperior** será el nombre distinguido primario. La información de contexto y todos los valores distinguidos alternativos estarán incluidos en el componente **valuesWithContext** del **AttributeTypeAndDistinguishedValue** de cualquier RDN, según se describe en 9.3 de la Rec. UIT-T X.501 | ISO/CEI 9594-2.

### 24.1.3 Iniciador

#### 24.1.3.1 Establecimiento

El establecimiento de una vinculación operacional jerárquica puede ser iniciado por cualquiera de los dos roles. La iniciación por el DSA superior puede ser causada por una operación añadir inserción con el DSA subordinado especificado en la extensión **targetSystem**, o por intervención administrativa. La iniciación por el DSA subordinado (que conecta una inserción o subárbol localmente existente al DIT global) es causada por intervención administrativa.

#### 24.1.3.2 Modificación

La modificación de una vinculación operacional jerárquica puede ser iniciada por cualquiera de los dos roles. El DSA superior puede emitir la modificación como resultado de una modificación de la información de prefijo de contexto superior. Éste a su vez puede ser el resultado de cualquiera de las operaciones de modificación, o de la intervención administrativa.

Cualquiera de los dos DSA puede modificar el acuerdo como resultado de una modificación del RDN de la inserción de prefijo de contexto del contexto de denominación subordinado. El DSA superior inicia esta modificación porque se está modificando un nombre distinguido relativo más arriba en el DIT. El DSA inicia esta modificación como consecuencia de una operación ModifyDN, o como consecuencia de una intervención administrativa.

Cualquiera de los DSA puede también modificar la HOB si la información de punto de acceso para su contexto de denominación cambia.

#### 24.1.3.3 Terminación

La terminación de una vinculación operacional jerárquica puede ser iniciada por cualquiera de los dos roles. La iniciación por el DSA superior puede ser causada por una intervención administrativa. La iniciación por el DSA subordinado puede ser causada bien por una operación suprimir inserción que suprime la inserción de prefijo de contexto del contexto de denominación subordinado, o bien por una intervención administrativa.

### 24.1.4 Parámetros de establecimiento

Los parámetros de establecimiento para los dos roles de una HOB, DSA superior y DSA subordinado, son diferentes. El parámetro de establecimiento para el rol DSA superior es un valor de **SuperiorToSubordinate**, en tanto que el parámetro para el rol subordinado es un valor de **SubordinateToSuperior**.

#### 24.1.4.1 Parámetro de establecimiento de DSA superior

El parámetro de establecimiento emitido por el DSA superior, un valor de **SuperiorToSubordinate**, proporciona al DSA subordinado información relativa a vértices del DIT superiores al prefijo de contexto del nuevo contexto de denominación (que incluye la referencia superior inmediata) y opcionalmente atributos de usuario y operacionales para la inserción de prefijo de contexto subordinada y copias de atributos de usuario y operacionales provenientes de la inserción inmediatamente superior al nuevo prefijo de contexto.

```

SuperiorToSubordinate ::= SEQUENCE {
    contextPrefixInfo      [0] DITcontext,
    entryInfo             [1] SET OF Attribute OPTIONAL,
    immediateSuperiorInfo [2] SET OF Attribute OPTIONAL }

```

El **rdn** en **Vertex** o en **SubentryInfo** será el RDN primario, y la información de contexto y todos los otros valores distinguidos estarán incluidos en los componentes **AttributeTypeAndDistinguishedValue** del RDN, según se indica en 9.3 de la Rec UIT-T X.501 | ISO/CEI 9594-2.

##### 24.1.4.1.1 Información de prefijo de contexto

El componente **contextPrefixInfo** de **SuperiorToSubordinate** es un valor de tipo **DITcontext**, el cual es una secuencia de valores **Vertex**.

```

DITcontext ::= SEQUENCE OF Vertex

```

```

Vertex ::= SEQUENCE {
    rdn [0] RelativeDistinguishedName,
    admPointInfo [1] SET OF Attribute OPTIONAL,
    subentries [2] SET OF SubentryInfo OPTIONAL,
    accessPoints [3] MasterAndShadowAccessPoints OPTIONAL }

```

El componente **contextPrefixInfo** es la secuencia de RDN que forma el nombre distinguido del superior inmediato del nuevo prefijo de contexto, estando cada RDN (dado por el componente **rdn**) acompañado opcionalmente de información adicional.

El componente opcional **admPointInfo** de un **Vertex** señala que el vértice del DIT es un punto administrativo y proporciona, por lo menos, su atributo operacional **administrativeRole**.

La información de subinserción asociada con un punto administrativo es proporcionada por el componente **subentries** de un **Vertex**, que es un conjunto de uno o más valores **SubentryInfo**. Cada valor **SubentryInfo** está compuesto del RDN de la subinserción (el componente **rdn**) y los atributos de la subinserción (el componente **info**).

```

SubentryInfo ::= SEQUENCE {
    rdn [0] RelativeDistinguishedName,
    info [1] SET OF Attribute }

```

El componente opcional **accessPoints** de un **Vertex** señala que el vértice corresponde al prefijo de contexto del contexto de denominación inmediatamente superior. El superior utiliza este componente para proporcionar al subordinado la información requerida para su referencia superior inmediata.

NOTA – El punto de acceso maestro dentro de **accessPoints** es el mismo que el pasado en el parámetro **accessPoint** de las operaciones Establecimiento y Modificación de vinculación operacional.

##### 24.1.4.1.2 Información de inserción

El componente opcional **entryInfo** de **SuperiorToSubordinate** es un conjunto de atributos que establece el contenido de la nueva inserción de prefijo de contexto.

##### 24.1.4.1.3 Información de inserción superior inmediata

El componente opcional **immediateSuperiorInfo** de **SuperiorToSubordinate** es una copia de un conjunto de atributos, en particular **objectClass** y **entryACI**, de la inserción inmediatamente superior al nuevo prefijo de contexto.

NOTA – Este componente puede utilizarlo el subordinado para optimizar la evaluación de una petición de listar que genera un **ListResult** vacío para un objeto de base que es la inserción inmediatamente superior al prefijo de contexto subordinado [véase la nota al inciso 2) de 19.3.1.2.2].

#### 24.1.4.2 Parámetro de establecimiento de DSA subordinado

El parámetro de establecimiento emitido por el DSA subordinado, un valor de **SubordinateToSuperior**, proporciona al DSA superior la información relativa al contexto de denominación subordinado.

```

SubordinateToSuperior ::= SEQUENCE {
    accessPoints [0] MasterAndShadowAccessPoints OPTIONAL,
    alias [1] BOOLEAN DEFAULT FALSE,
    entryInfo [2] SET OF Attribute OPTIONAL,
    subentries [3] SET OF SubentryInfo OPTIONAL }

```

El componente **accessPoints** de **SubordinateToSuperior** lo utiliza el subordinado para proporcionar al superior la información requerida para su referencia subordinada.

NOTA 1 – El punto de acceso maestro dentro de **accessPoints** es el mismo que el pasado en el parámetro **accessPoint** de las operaciones Establecimiento y Modificación de vinculación operacional.

El componente **alias** de **SubordinateToSuperior** se utiliza para señalar al superior que el contexto de denominación subordinado está constituido por una sola inserción de alias.

El componente **entryInfo** de **SubordinateToSuperior** consiste en una copia de un conjunto de atributos, en particular **objectClass** y **entryACI**, de la nueva inserción de prefijo de contexto.

El componente **subentries** de **SubordinateToSuperior** lo utiliza el subordinado para pasar subinserciones que contienen ACI prescriptivas al superior.

NOTA 2 – El superior puede utilizar estos dos últimos componentes para optimizar la evaluación de una petición de listar o buscar a un nivel, cuyo objeto de base es la inserción inmediatamente superior al prefijo de contexto subordinado.

#### 24.1.5 Parámetros de modificación

Para modificaciones de una HOB, el parámetro de modificación del rol superior, **SuperiorToSubordinateModification**, es **SuperiorToSubordinate**, con la restricción de que el componente **entryInfo** no puede estar presente; el correspondiente al rol subordinado es **SubordinateToSuperior**.

```

SuperiorToSubordinateModification ::= SuperiorToSubordinate (
    WITH COMPONENTS { ..., entryInfo ABSENT})

```

Estos parámetros son idénticos (con la restricción antes mencionada) a los correspondientes parámetros de establecimiento y se utilizan para señalar cambios que tienen lugar en la información proporcionada en los parámetros de establecimiento subsiguientes al establecimiento de la HOB.

Si cualquier componente de **SuperiorToSubordinate** (o, subsiguientemente, **SuperiorToSubordinateModification**) o **SubordinateToSuperior** sufre un cambio (por ejemplo, el componente **contextPrefixInfo** de **SuperiorToSubordinate**), el componente correspondiente del parámetro de modificación (por ejemplo, el componente **contextPrefixInfo** de **SuperiorToSubordinateModification**) deberá ser proporcionado, íntegramente, en la vinculación operacional modificar.

#### 24.1.6 Parámetros de terminación

Ninguno de los dos roles proporciona un parámetro de terminación cuando termina una HOB.

#### 24.1.7 Identificación de tipo

La vinculación operacional jerárquica es identificada por el identificador de objeto asignado cuando se define el objeto de información **hierarchicalOperationalBinding OPERATIONAL-BINDING** en 24.2.

### 24.2 Definición de la clase de objeto de información vinculación operacional

Esta subcláusula define el tipo vinculación operacional jerárquica utilizando la plantilla de clase de objeto de información **OPERATIONAL-BINDING** definida en la Rec. UIT-T X.501 | ISO/CEI 9594-2.

```

hierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
    AGREEMENT HierarchicalAgreement
    APPLICATION CONTEXTS {
        {directorySystemAC} }
    ASYMMETRIC
        ROLE-A { -- superior DSA
            ESTABLISHMENT-INITIATOR TRUE
            ESTABLISHMENT-PARAMETER SuperiorToSubordinate
            MODIFICATION-INITIATOR TRUE
            MODIFICATION-PARAMETER SuperiorToSubordinateModification
            TERMINATION-INITIATOR TRUE }

```

```

ROLE-B {          -- subordinate DSA
    ESTABLISHMENT-INITIATOR          TRUE
    ESTABLISHMENT-PARAMETER          SubordinateToSuperior
    MODIFICATION-INITIATOR           TRUE
    MODIFICATION-PARAMETER           SubordinateToSuperior
    TERMINATION-INITIATOR             TRUE }
ID                                     id-op-binding-hierarchical }

```

### 24.3 Procedimientos DSA para gestión de vinculaciones operacionales

En los procedimientos siguientes, un nuevo DSE, o una marca (es decir, una indicación de estado asociada con algún ítem de información) creado por un DSA deberá almacenarse en un dispositivo de almacenamiento estable. Al hacer así, los dos DSA sigan los procedimientos descritos más abajo podrán mantener una comprensión consistente de los parámetros de la HOB en presencia de fallos de la comunicación y de un sistema de extremo.

Tanto en el procedimiento de **establecimiento** como en el de **modificación** descritos más abajo, el DSA que desempeña el rol respondedor (es decir, que no inicia el establecimiento o la modificación) podría proporcionar al DSA que desempeña el rol iniciador unas informaciones (por ejemplo, atributos operacionales) que no son aceptables por una razón u otra. El DSA iniciador puede terminar la vinculación operacional en tales casos.

#### 24.3.1 Procedimiento de establecimiento

##### 24.3.1.1 Establecimiento iniciado por el DSA superior

Si un DSA evalúa una operación de añadir inserción con un DSA diferente especificado en la extensión **targetSystem**, deberá establecer una vinculación operacional jerárquica de acuerdo con el siguiente procedimiento. Si un DSA, por razones administrativas, desea establecer una HOB con un DSA subordinado, y él soporta el protocolo DOP HOB, deberá seguirse el siguiente procedimiento:

- 1) El DSA superior crea una nueva DSE de tipo **subr**, con el nombre de la nueva inserción, y marca esta nueva DSE como *siendo añadida*. El DSA genera una **bindingID** única y la almacena con la nueva DSE.
- 2) El DSA superior enviará al DSA subordinado una operación vinculación operacional de establecer que contiene los siguientes parámetros:
  - a) **bindingType** fijado a **hierarchicalOperationalBindingID**,
  - b) el parámetro de establecimiento **SuperiorToSubordinate** con los componentes **contextPrefixInfo** y **entryInfo** presentes; todos los demás parámetros son opcionales,
  - c) **HierarchicalAgreement** con el componente **immediateSuperior** fijado al nombre distinguido del superior inmediato de la nueva inserción y el componente **rdn** fijado al RDN de la nueva inserción,
  - d) la **bindingID**, **myAccessPoint** y parámetros **válidos**, según proceda.
- 3) Si el DSA subordinado acepta la operación, crea las DSE requeridas de tipos **glue**, **subentry**, **admPoint**, **rhob** e **immSupr**, según proceda, para representar la **contextPrefixInfo**; una DSE de tipo **cp** y **entry** o **alias** para representar la nueva inserción de objeto de prefijo o inserción de alias; y, según proceda, una DSE de tipo **rhob** y **entry** para representar la **immediateSuperiorInfo**. Almacena la **bindingID** con la DSE de la nueva inserción de prefijo de contexto y retorna un parámetro **SubordinateToSuperior** al DSA superior.

Si el DSA subordinado rehúsa la operación, devuelve un error de vinculación operacional con el valor de problema apropiado fijado.

Si el contexto de denominación ya existe y los valores **bindingID** para el contexto existente y el nuevo son los mismos, el DSA subordinado ya ha creado el contexto de denominación solicitado, en cuyo caso el DSA subordinado retorna un resultado al superior. Si los valores no son iguales, se envía un error de vinculación operacional con el problema **invalidAgreement**; esto significa que el DSA superior tiene una inconsistencia de conocimiento permanente que requiere corrección por un administrador.

- 4) Si el DSA superior recibe un error, suprime la DSE marcada de tipo **subr** y retorna un error para la operación de añadir inserción.

Si el DSA superior recibe un resultado, suprime la marca de la DSE que representa el **subr** y retorna un resultado para la operación añadir inserción.

Si se produce un fallo cualquiera (por ejemplo, un fallo de la comunicación o de un sistema de extremo), el DSA superior repetirá los pasos comenzando por el paso 2) hasta que haya recibido un resultado o un error para cada establecimiento pendiente de una vinculación operacional jerárquica de la cual él es el iniciador. Si el establecimiento es el resultado de una operación de añadir inserción, y el peticionario aborta la operación (por ejemplo, liberando o abortando la asociación de aplicación) antes de que el establecimiento haya sido concluido, el DSA superior ignorará este suceso y concluirá establecimiento (que podrá o no tener éxito). En este caso, el usuario no será informado del resultado final de la operación añadir inserción.

NOTA 1 – El marcaje por el subordinado ayuda al control de recuperación y concurrencia. Otro usuario no puede añadir una inserción que esté ya marcada, y, tras un fallo, el DSA repite la vinculación operacional de establecer para todos los subordinados marcados.

NOTA 2 – Con el procedimiento antes descrito, el conocimiento sólo tiene una inconsistencia transitoria. Es asunto local determinar cómo el DSA superior trata operaciones no relacionadas que leen la referencia subordinada mientras está marcada.

#### 24.3.1.2 Establecimiento iniciado por un DSA subordinado

El DSA subordinado puede iniciar una vinculación operacional jerárquica. Ésta podría producirse cuando un administrador desea conectar un subárbol de inserciones contenida en el DSA a cierto punto en el DIT global. En este caso, el DSA subordinado deberá establecer una HOB de acuerdo con el siguiente procedimiento:

- 1) El DSA subordinado, o bien tiene una DSE de tipo **cp** como una parte de un contexto de denominación existente, o bien crea una nueva. El DSA marca la DSE como *siendo añadida*, genera una **bindingID** única, y la almacena con la DSE de prefijo de contexto.
- 2) El DSA subordinado envía al DSA superior una operación de vinculación operacional de establecer que contiene los siguientes parámetros:
  - a) **bindingType** puesto a **hierarchicalOperational BindingID**,
  - b) el parámetro de establecimiento **SubordinateToSuperior**, según proceda,
  - c) **HierarchicalAgreement** con el componente **immediateSuperior** puesto al nombre distinguido del superior inmediato de la nueva inserción y el componente **rdn** puesto al RDN de la nueva inserción,
  - d) **bindingID**, **myAccessPoint** y parámetros válidos, según proceda.

Si el DSA superior rehúsa la operación, retorna un error de vinculación operacional con el valor apropiado de problema fijado.

- 3) El DSA superior comprueba que él es maestro para el superior inmediato de la nueva inserción de prefijo de contexto; si no lo es, retorna un error de vinculación operacional con el problema **roleAssignment**.
- 4) El DSA superior comprueba que el RDN solicitado para el nuevo prefijo de contexto no está ya en uso. Si no se encuentra un RDN concordante utilizando información contenida localmente, pero la DSE inmediatamente superior es de tipo **nssr**, se sigue el procedimiento de 19.1.5. Si no se descubre un RDN concordante utilizando este procedimiento, el DSA superior crea una DSE de tipo **subr**, almacena la **bindingID** con dicha DSE, y retorna un resultado.

Si se encuentra una referencia subordinada con este RDN, se comparan los dos valores de **bindingID**. Si son iguales, se retorna un resultado. El parámetro **SuperiorToSubordinate** retornado por el DSA superior no contendrá el componente **entry**. Si los dos valores de **bindingID** no son iguales, se envía un error de vinculación operacional con el problema **invalidAgreement**; esto significa que el DSA superior tiene una inconsistencia de conocimiento permanente y requiere corrección por un administrador.

Si al explorar una NSSR se encuentra un RDN concordante, se envía un error de vinculación operacional con el problema **invalidAgreement**. Esto significa que el DSA superior tiene una inconsistencia de conocimiento permanente que requiere corrección por un administrador.

- 5) Si el DSA subordinado recibe un error, suprime la nueva DSE de prefijo de contexto y su marca. Es un asunto local determinar el destino que habrá de darse a la información de inserción de la cual se derivó la DSE de prefijo de contexto.

Si el DSA subordinado recibe un resultado, añade las DSE necesarias de tipos **glue**, **subentry**, **admPoint**, **rhob** e **immSupr**, según proceda, para representar la **contextPrefixInfo** y, según proceda, un DSE de tipo **rhob** y **entry** para representar la **immediateSuperiorInfo**. Se suprime la marca del DSE de prefijo de contexto.

Si se produce un fallo cualquiera (por ejemplo, de la comunicación o de un sistema de extremo), el DSA subordinado repetirá los pasos comenzando por el paso 2) hasta que se haya recibido un resultado para cada establecimiento pendiente de una vinculación operacional jerárquica de la cual él es el iniciador.



### 24.3.2 Procedimiento de modificación

Los siguientes procedimientos se definen para la modificación de una HOB que ya ha sido iniciada por el procedimiento descrito en 24.3.1.

#### 24.3.2.1 Procedimiento de modificación iniciado por un DSA superior

Este procedimiento puede ser invocado como resultado de operaciones de modificación, como se indica en 19.1, o como resultado de una intervención administrativa (por ejemplo, para introducir cambios en los parámetros **myAccessPoint**, **agreement** o **valid** de la HOB). Asimismo, si un DSA superior detecta cambios introducidos en los componentes **contextPrefixInfo** o **immediateSuperiorInfo** del valor **SuperiorToSubordinate** que se suministra al DSA subordinado, propagará la nueva información al DSA subordinado empleando el siguiente procedimiento:

- 1) Se marca la DSE de tipo **subr** como *siendo modificada*, y, si este tipo de modificación es el resultado de una modificación del RDN de la inserción de prefijo de contexto subordinado, se añade una nueva DSE de tipo **subr** la cual se marca como *siendo añadida*.
- 2) El DSA superior produce un nuevo valor **bindingID** a partir del valor existente incrementando su componente **version**. Utilizando esta nueva **bindingID**, envía una operación de vinculación operacional de modificar al DSA subordinado con el parámetro de modificación **SuperiorToSubordinateModification**.
- 3) El DSA subordinado comprueba el componente **identifier** de la **bindingID**. Si no tiene tal acuerdo con el superior, o si el componente **version** es menor que la versión de la HOB, deberá retornar un error de vinculación operacional con el problema **invalidAgreement**.
- 4) El DSA subordinado puede aceptar la modificación de la HOB, modificar o reconstruir las DSE que representan la información de prefijo de contexto, actualizar el componente **version** de su **bindingID** y retornar un resultado. Como una alternativa, puede retornar un error y, después de esto, terminar el acuerdo.
- 5) Si el DSA superior recibe un resultado, la modificación está concluida. Si esta modificación es consecuencia de una modificación del RDN de la inserción de prefijo de contexto subordinado, se suprime la marca de la nueva DSE que tiene el tipo **subr** y ha sido marcada como *siendo añadida*, y se suprime la antigua DSE marcada como *siendo modificada*. De no ser así, lo único que se hace es suprimir la marca *siendo modificada*.

Si el DSA superior recibe un error, la modificación ha fracasado. Se suprime la marca *siendo modificada*. Si esta modificación es el resultado de una modificación del RDN de la inserción de prefijo de contexto subordinado, se suprime la nueva DSE que tiene el tipo **subr** y está marcada como *siendo añadida*. En otro caso, las medidas adoptadas quedan fuera del ámbito de esta Especificación de directorio.

Si se produce un fallo cualquiera (por ejemplo de la comunicación o de un sistema de extremo), el DSA superior repetirá los pasos comenzando por el paso 2) hasta que se haya recibido un resultado o un error para cada modificación pendiente, de una vinculación operacional jerárquica de la cual él es el iniciador. Si la modificación es el resultado de una operación ModifyDN que modifica el RDN de la inserción de prefijo de contexto subordinado, y el peticionario aborta la operación (por ejemplo, liberando o abortando la asociación de aplicación) antes de que haya sido concluida la modificación, el DSA superior ignorará este suceso y concluirá la modificación (que podrá o no tener éxito). En este caso, el usuario no será informado del resultado final de la operación ModifyDN.

#### 24.3.2.2 Procedimiento de modificación iniciado por un DSA subordinado

Este procedimiento puede ser invocado como resultado de una intervención administrativa (por ejemplo, para introducir cambios en los parámetros **myAccessPoint**, **agreement** o **valid** de la HOB). Asimismo, si un DSA subordinado detecta cambios introducidos en el valor **SubordinateToSuperior** que suministró al DSA superior, propagará la nueva información al DSA superior empleando el siguiente procedimiento:

- 1) Marcará la DSE de tipo **cp** como *siendo modificada*.
- 2) El DSA subordinado produce un nuevo valor **bindingID** a partir del valor existente incrementando su componente **version**. Utilizando esta nueva **bindingID**, envía una operación de vinculación operacional de modificar al DSA superior con el parámetro de modificación **SubordinateToSuperior**.
- 3) El DSA superior comprueba el componente **identifier** de la **bindingID**. Si no tiene tal acuerdo con el subordinado, o si el componente **version** es menor que la versión de la HOB, retornará un error de vinculación operacional con el problema **invalidAgreement**.
- 4) El DSA superior puede aceptar la modificación de la HOB, modificar la DSE que representa la referencia subordinada y retornar un resultado. Como otra posibilidad, puede retornar un error y, después de esto, terminar el acuerdo.

Además, si la DSE superior de la DSE (de tipo **subr**) que va a ser redeterminada es de tipo **nssr**, el DSA seguirá el procedimiento definido en 19.1.5 (operaciones de modificar y referencias subordinadas no específicas) para asegurar que el nuevo nombre de la inserción es inequívoco, antes de responder a la petición de modificación de HOB.

- 5) Si el DSA subordinado recibe un resultado, la modificación está concluida; el DSA y suprime entonces la marca. Las medidas que habrá que tomar en el caso de que reciba un error quedan fuera del ámbito de esta Especificación de directorio.

Si se produce cualquier fallo (por ejemplo de la comunicación o de un sistema de extremo), el DSA subordinado repetirá los pasos comenzando por el paso 2) hasta que haya recibido un resultado para cada modificación de una vinculación operacional jerárquica que esté pendiente y de la cual él es el iniciador.

### 24.3.3 Procedimiento de terminación

Se han definido los siguientes procedimientos para la terminación de una HOB que ha sido iniciada por el procedimiento descrito en 24.3.1.

#### 24.3.3.1 Terminación iniciada por el DSA superior

La terminación de una vinculación operacional jerárquica es iniciada por el DSA superior solamente como resultado de una intervención administrativa. Deberá emplearse el siguiente procedimiento:

- 1) El DSA superior marca la DSE que representa la referencia subordinada como *siendo suprimida*, de modo que la referencia subordinada no se utilice más durante la resolución de nombre.
- 2) El DSA superior envía al DSA subordinado una operación de vinculación operacional de terminar para la vinculación operacional jerárquica. El superior omite el componente **version** de **bindingID**.
- 3) Cuando el DSA subordinado recibe la vinculación operacional de terminar, suprime cualquier información sobre la vinculación operacional jerárquica y envía un resultado, a menos que el componente **identif** de la **bindingID** sea desconocido, en cuyo caso retorna un error de vinculación operacional con el problema **invalidID**. Es un asunto local determinar el destino que ha de darse a la información de inserción asociada con el contexto de denominación subordinado.
- 4) Si el DSA superior recibe un resultado o un error de vinculación operacional con el problema **invalidID**, suprimirá la DSE marcada como *siendo suprimida*, que representa la referencia subordinada asociada con la vinculación operacional jerárquica y suprime cualquier información sobre la vinculación operacional.

Si se produce cualquier error (por ejemplo, de la comunicación o de un sistema de extremo) el DSA superior repetirá los pasos comenzando por el paso 2) hasta que se haya recibido un resultado para cada terminación pendiente de una vinculación operacional jerárquica del cual él es el iniciador.

#### 24.3.3.2 Terminación iniciada por el DSA subordinado

Una terminación iniciada por el DSA subordinado puede ser causada por una operación de suprimir inserción que suprime la última inserción dentro del contexto de denominación subordinado, la inserción de prefijo de contexto, o como resultado de una intervención administrativa. Deberá seguirse el siguiente procedimiento:

- 1) El DSA subordinado marca la DSE de prefijo de contexto del contexto de nominación como *siendo suprimida*.
- 2) El DSA envía al DSA superior una operación de vinculación operacional de terminar para la vinculación operacional jerárquica. El DSA subordinado omite el componente **version** de la **bindingID**.
- 3) Cuando el DSA superior recibe la vinculación operacional de terminar, suprime la DSE que representa la referencia subordinada asociada con la vinculación operacional jerárquica, suprime toda información sobre la vinculación operacional y envía un resultado, a menos que el componente **identif** de la **bindingID** sea desconocido, en cuyo caso retorna un error de vinculación operacional con el problema **invalidID**.
- 4) Si el DSA subordinado recibe un resultado o un error de vinculación operacional con el problema **invalidID**, suprimirá toda información sobre la vinculación operacional.

NOTA – El destino que ha de darse a la información de inserción de un contexto de nominación es un asunto local del DSA subordinado. Puesto que la redeterminación (o lo que viene a ser lo mismo, el traslado) de un contexto de denominación no está autorizado por la operación ModifyDN, un administrador pudiera, por ejemplo, terminar la HOB, seleccionar otro prefijo de contexto para el contexto de denominación, y reconectarlo a otra parte del DIT (es decir, establecer una nueva HOB).

Si se produce cualquier fallo (por ejemplo, de la comunicación o de un sistema de extremo), el DSA subordinado repetirá los pasos comenzando por el paso 2) hasta que se haya recibido un resultado para cada terminación de una vinculación operacional jerárquica que esté pendiente y de la cual él es el iniciador.

## 24.4 Procedimientos para operaciones

Las operaciones que pueden ser ejecutadas en el estado cooperativo de una vinculación operacional jerárquica son los definidos dentro del contexto de aplicación **directorySystemAC**.

Los procedimientos que deberá seguir el DSA que interviene en una vinculación operacional jerárquica se definen en las cláusulas 16 a 22.

## 24.5 Utilización de contextos de aplicación

Para establecer, modificar o terminar una vinculación operacional jerárquica utilizando el protocolo y los procedimientos de esta Especificación de directorio, un DSA deberá utilizar el contexto de aplicación **operationalBindingManagementAC**.

## 25 Vinculación operacional jerárquica no específica

Se utiliza una vinculación operacional jerárquica no específica para representar la relación entre dos DSA que contienen dos contextos de denominación, uno inmediatamente subordinado al otro. En el caso de una NHOB, el DSA superior contiene una referencia subordinada no específica al contexto de denominación contenido por el DSA subordinado; el DSA subordinado contiene una referencia superior inmediata al contexto de denominación contenido por el DSA superior. La vinculación operacional asegura que la información de conocimiento apropiada sea intercambiada y mantenida entre los dos DSA de modo que ambos puedan comportarse durante el proceso de resolución de nombre y la evaluación de operación tal como está definido en las cláusulas 18 y 19.

### 25.1 Características del tipo vinculación operacional

#### 25.1.1 Simetría y roles

El tipo vinculación operacional jerárquica es un tipo asimétrico de vinculación operacional. Los dos roles de este tipo son:

- a) el rol del DSA maestro para el contexto de denominación superior, el *DSA superior* (asociado con el rol abstracto "A"); y
- b) el rol del DSA maestro para el contexto de denominación subordinado, el *DSA subordinado* (asociado con el rol abstracto "B").

#### 25.1.2 Acuerdo

La información de acuerdo intercambiada durante el establecimiento de la vinculación operacional jerárquica no específica, un valor de **NonSpecificHierarchicalAgreement**, contiene solamente el nombre distinguido del superior inmediato del nuevo contexto de denominación (el componente **immediateSuperior**). Esta información será proporcionada por el DSA que inicia la NHOB.

```
NonSpecificHierarchicalAgreement ::= SEQUENCE {  
    immediateSuperior [1] DistinguishedName }
```

NOTA – La manera en que el DSA subordinado determina que el nombre del nuevo contexto de denominación es inequívoco queda fuera del ámbito de esta Recomendación | Norma Internacional. El nombre será inequívoco si es correctamente asignado por la autoridad de denominación apropiada y si ningún otro DSA contiene el mismo nombre como una inserción maestra.

#### 25.1.3 Iniciador

##### 25.1.3.1 Establecimiento

El establecimiento de una vinculación operacional jerárquica no específica puede ser iniciado solamente por el rol DSA subordinado. La iniciación por el DSA subordinado (que conecta una o más inserciones o árboles localmente existentes al DIT global) es causada por una intervención administrativa.

### 25.1.3.2 Modificación

La modificación de una vinculación jerárquica no específica puede ser iniciada por cualquiera de los dos roles. El DSA superior puede emitir la modificación como resultado de una modificación de la información de prefijo de contexto superior. Ésta puede a su vez ser el resultado de cualesquiera operaciones de modificación, o de una intervención administrativa.

Cualquiera de los dos DSA puede también modificar la NHOB si la información de punto de acceso para su contexto de denominación (o uno de sus contextos de denominación inmediatamente subordinados en el caso del rol subordinado) cambia.

### 25.1.3.3 Terminación

La terminación de una vinculación operacional jerárquica puede ser iniciada por cualquiera de los dos roles. La iniciación por el DSA superior puede ser causada por una intervención administrativa. La iniciación por el DSA subordinado puede ser causada o bien por la operación suprimir inserción que suprime la inserción de prefijo de contexto final contenido por el subordinado inmediatamente subordinado al componente **immediateSuperior** del acuerdo o por una intervención administrativa.

### 25.1.4 Parámetros de establecimiento

El parámetro de establecimiento emitido por el DSA superior, un valor de **NHOBSuperiorToSubordinate**, es equivalente al correspondiente parámetro de establecimiento HOB, excepto que el componente **entryInfo** está ausente.

**NHOBSuperiorToSubordinate ::= SuperiorToSubordinate (**  
**WITH COMPONENTS { ..., entryInfo ABSENT})**

El parámetro de establecimiento emitido por el DSA subordinado, un valor de **NHOBSubordinateToSuperior**, es equivalente al correspondiente parámetro de establecimiento HOB, excepto que los componentes **alias** y **entryInfo** están ausentes.

**NHOBSubordinateToSuperior ::= SubordinateToSuperior (**  
**WITH COMPONENTS { ..., alias ABSENT, entryInfo ABSENT})**

### 25.1.5 Parámetros de modificación

Estos parámetros son idénticos a los correspondientes parámetros de establecimiento y se utilizan para señalar cambios que se producen en la información proporcionada en los parámetros de establecimiento subsiguientes al establecimiento de la NHOB.

Si cualquier componente de **NHOBSuperiorToSubordinate** o **NHOBSubordinateToSuperior** experimenta un cambio (por ejemplo, el componente **contextPrefixInfo** de **NHOBSuperiorToSubordinate**), el componente correspondiente del parámetro de modificación (por ejemplo, el **contextPrefixInfo** de **NHOBSuperiorToSubordinate**) será proporcionado íntegramente en la vinculación operacional de modificar.

### 25.1.6 Parámetros de terminación

Ninguno de los dos roles proporciona un parámetro de terminación cuando termina una NHOB.

### 25.1.7 Identificación de tipo

La vinculación operacional jerárquica no específica es identificada por el identificador de objeto asignado cuando se define el objeto de información **nonSpecificHierarchicalOperationalBinding OPERATIONAL-BINDING** en 25.2.

## 25.2 Definición de la clase de objeto de información vinculación operacional

Esta subcláusula define el tipo de vinculación operacional jerárquica no específica utilizando la plantilla de clase de objeto de información **OPERATIONAL-BINDING** definida en la Rec. UIT-T X.501 | ISO/CEI 9594-2.

```

nonSpecificHierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
  AGREEMENT      NonSpecificHierarchicalAgreement
  APPLICATION CONTEXTS {
    { directorySystemAC } }
  ASYMMETRIC
    ROLE-A {      -- superior DSA
      ESTABLISHMENT-PARAMETER  NHOBSuperiorToSubordinate
      MODIFICATION-INITIATOR    TRUE
      MODIFICATION-PARAMETER    NHOBSuperiorToSubordinate
      TERMINATION-INITIATOR     TRUE }
    ROLE-B {      -- subordinate DSA
      ESTABLISHMENT-INITIATOR    TRUE
      ESTABLISHMENT-PARAMETER    NHOBSubordinateToSuperior
      MODIFICATION-INITIATOR    TRUE
      MODIFICATION-PARAMETER    NHOBSubordinateToSuperior
      TERMINATION-INITIATOR    TRUE }
  ID              id-op-binding-non-specific-hierarchical }

```

### 25.3 Procedimientos de DSA para gestión de vinculaciones operacionales jerárquicas no específicas

En los siguientes procedimientos, al igual que en los descritos en 24.3, una nueva DSE o una marca creadas por un DSA deberán ser almacenadas en un dispositivo de almacenamiento estable.

Tanto en el procedimiento de establecimiento como en el de modificación descritos más abajo, el DSA que desempeña el rol respondedor (es decir, el que no inicia el establecimiento o la modificación) puede proporcionar al DSA que desempeña el rol de iniciador informaciones (por ejemplo atributos operacionales) que no son aceptables por una razón u otra. El DSA iniciador puede terminar la vinculación operacional en tales casos.

#### 25.3.1 Procedimiento de establecimiento

Solamente el DSA subordinado puede iniciar una vinculación operacional jerárquica. Esta podría producirse cuando un administrador desea conectar uno o más subárboles de inserciones contenidas en el DSA a cierto punto en el DIT global. En este caso, el DSA subordinado establecerá una NHOB de acuerdo con el siguiente procedimiento:

- 1) El DSA subordinado, o bien tiene una DSE de tipo **cp** como parte de un contexto de denominación existente, o crea una nueva. El DSA marca la DSE como *siendo añadida*, y genera una **bindingID** única, y la almacena con la DSE de prefijo de contexto.
- 2) El DSA subordinado envía al DSA superior una operación de vinculación operacional de establecer que contiene los siguientes parámetros:
  - a) **bindingType** fijado a **nonSpecificHierarchicalOperationalBindingID**,
  - b) el parámetro de establecimiento **NHOBSubordinateToSuperior**, según proceda,
  - c) **NonSpecificHierarchicalAgreement** con el componente **immediateSuperior** fijado al nombre distinguido del superior inmediato de la nueva inserción,
  - d) los parámetros **bindingID**, **myAccessPoint** y **valid**, según proceda.
- 3) El DSA superior comprueba que él es maestro para el superior inmediato de la nueva inserción de prefijo de contexto; de no serlo, retorna un error de vinculación operacional con el problema **roleAssignment**.
- 4) El DSA superior añade la DSE de tipo **nssr** (e información de atributo **nonSpecificKnowledge**) a la DSE del superior inmediato de la nueva inserción, almacena la **bindingID** con esa DSE, y retorna un resultado.
- 5) Si el DSA subordinado recibe un error, suprime la nueva DSE de prefijo de contexto y su marca. Es un asunto local determinar el destino que ha de darse a la información de la cual se derivó la DSE de contexto de prefijo.

Si el DSA subordinado recibe un resultado, añade las DSE necesarias de tipos **glue**, **subentry**, **admPoint**, **rhob** e **immSupr**, según proceda, para representar la **contextPrefixInfo**, y, si procede, una DSE de tipo **rhob** y **entry** para representar la **immediateSuperiorInfo**. Se suprime la marca de la DSE de prefijo de contexto.

Si se produce cualquier fallo (por ejemplo en la comunicación o en un sistema de extremo), el DSA subordinado repetirá los pasos comenzando por el paso 2) hasta que se haya recibido un resultado para cada establecimiento pendiente de una vinculación operacional jerárquica de la cual él es el iniciador.

### 25.3.2 Procedimiento de modificación

Si el DSA superior detecta cualquier cambio en la información **NHOBSuperiorToSubordinate** que haya sido suministrada a un DSA subordinado dentro de una vinculación operacional jerárquica no específica, propagará la información modificada al DSA subordinado. Si la NHOB se estableció utilizando los procedimientos de 25.3.1, deberá modificarse de acuerdo con los procedimientos definidos para modificar la vinculación operacional jerárquica en 24.3.2.1 (con **NHOBSuperiorToSubordinate** en sustitución de **SuperiorToSubordinateModification**).

De manera similar, si el DSA subordinado detecta cualquier cambio en la información **NHOBSubordinateToSuperior** que suministró a un DSA superior, propagará los cambios al DSA superior. Si la NHOB se estableció utilizando los procedimientos de 25.3.1, deberá modificarse de acuerdo con los procedimientos definidos para modificar la vinculación operacional jerárquica en 24.3.2.2 (con **NHOBSubordinateToSuperior** en sustitución de **SubordinateToSuperior**).

### 25.3.3 Procedimiento de terminación

Los siguientes procedimientos se definen para la terminación de una NHOB establecida mediante los procedimientos de 25.3.1.

#### 25.3.3.1 Terminación iniciada por un DSA superior

La terminación de una vinculación operacional jerárquica la inicia el DSA superior solamente como resultado de una intervención administrativa. Deberá seguirse el siguiente procedimiento:

- 1) El DSA superior marca el valor correspondiente al DSA subordinado en el atributo **nonSpecificKnowledge** contenido en la DSE de la inserción inmediatamente superior, como *siendo suprimida*.
- 2) El DSA superior envía una operación de vinculación operacional de terminar para la NHOB con el DSA subordinado. El superior omite el componente **version** de la **bindingID**.
- 3) Cuando el DSA subordinado recibe la vinculación operacional de terminar, suprime cualquier información sobre la NHOB y envía un resultado, a menos que el componente **identifier** de la **bindingID** sea desconocido, en cuyo caso retorna un error de vinculación operacional con el problema **invalidID**. Es un asunto local determinar el destino que ha de darse a la información de inserción asociada con el contexto de denominación subordinado.
- 4) Si el DSA superior recibe un resultado o un error de vinculación operacional con el problema **invalidID**, suprimirá el valor del atributo **nonSpecificKnowledge** marcado *siendo suprimida* que representa la información de punto de acceso asociada con la NHOB y suprime cualquier información sobre la vinculación operacional. Si éste fue el último valor del atributo **nonSpecificKnowledge**, suprime este atributo **nonSpecificKnowledge** y la DSE de tipo **nssr** en la DSE.

Si se produce cualquier fallo (por ejemplo, en la comunicación o en un sistema de extremo), el DSA superior repetirá los pasos comenzando por el paso 2) hasta que se haya recibido un resultado de error para cada terminación pendiente de una NHOB de la cual él es el iniciador.

#### 25.3.3.2 Terminación iniciada por el DSA subordinado

La terminación iniciada por el DSA subordinado puede ser causada por una operación de suprimir inserción que suprime la última inserción dentro del contexto de denominación subordinado, la inserción de contexto de prefijo, del último contexto de denominación subordinado contenido por el DSA subordinado, o causada por una intervención administrativa. Se sigue el siguiente procedimiento:

- 1) El DSA subordinado marca la DSE de prefijo de contexto del contexto de denominación como *siendo suprimida*.
- 2) El DSA subordinado envía al DSA superior una operación de vinculación operacional de terminar para la vinculación operacional jerárquica. El subordinado omite el componente **version** de la **bindingID**.
- 3) Cuando el DSA superior recibe la vinculación operacional de terminar, suprime el valor del atributo **nonSpecificKnowledge** que representa la información de punto de acceso asociada con la NHOB, borra toda información sobre la vinculación operacional, suprime el atributo **nonSpecificKnowledge** y la DSE de tipo **nssr** en la DSE inmediatamente superior al contexto de denominación subordinado (si el valor suprimido fue el último valor del atributo **nonSpecificKnowledge**) y envía un resultado, a menos que el componente **identifier** de la **bindingID** sea desconocido, en cuyo caso retorna un error de vinculación operacional con el problema **invalidID**.

- 4) Si el DSA subordinado recibe un resultado o un error de vinculación operacional con el problema **invalidID**, deberá suprimir toda información sobre la vinculación operacional. Es un asunto local determinar el destino que ha de darse a toda información de inserción asociada con el contexto de denominación subordinado.

Si se produce cualquier error (por ejemplo, en la comunicación o en un sistema de extremo), el DSA subordinado repetirá los pasos comenzando por el paso 2) hasta que se haya recibido un resultado o error para cada terminación pendiente de una NHOB de la cual él es el iniciador.

#### 25.4 Procedimientos para operaciones

Las operaciones que pueden ser ejecutadas en el estado cooperativo de una vinculación operacional jerárquica no específica son los definidos dentro del contexto de aplicación **directorySystemAC**.

Los procedimientos que deberá seguir el DSA que interviene en una vinculación operacional jerárquica no específica serán los definidos en las cláusulas 16 a 22.

#### 25.5 Utilización de contextos de aplicación

Para establecer, modificar o terminar una vinculación operacional jerárquica no específica utilizando el protocolo y los procedimientos de esta Especificación de directorio, un DSA deberá utilizar el contexto de aplicación **operationalBindingManagementAC**.

## Anexo A

## ASN.1 para operaciones distribuidas

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

Este anexo incluye todas las definiciones de tipo, valor y macro ASN.1 contenidas en esta Especificación de directorio en la forma del módulo ASN.1 **DistributedOperations**.

**DistributedOperations** {joint-iso-itu-t ds(5) module(1) distributedOperations(3) 3}

**DEFINITIONS ::=**

**BEGIN**

**-- EXPORTS All --**

*-- The types and values defined in this module are exported for use in the other ASN.1 modules contained within the Directory Specifications, and for the use of other applications which will use them to access Directory services. Other applications may use them for their own purposes, but this will not constrain extensions and modifications needed to maintain or improve the Directory service.*

**IMPORTS**

**informationFramework, directoryAbstractService, distributedOperations, selectedAttributeTypes, basicAccessControl, dap**  
**FROM UsefulDefinitions** {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 3}

**DistinguishedName, Name, RDNSequence**  
**FROM InformationFramework** informationFramework

**PresentationAddress, ProtocolInformation, UniqueIdentifier**  
**FROM SelectedAttributeTypes** selectedAttributeTypes

**AuthenticationLevel**  
**FROM BasicAccessControl** basicAccessControl

**OPERATION, ERROR**  
**FROM Remote-Operations-Information-Objects**  
 {joint-iso-ccitt remote-operations(4) informationObjects(5) version1(0)}

**directoryBind, directoryUnbind, read, compare, abandon, list, search, addEntry, removeEntry, modifyEntry, modifyDN, referral, SecurityParameters, CommonResults**  
**FROM DirectoryAbstractService** directoryAbstractService

**DIRQOP, OPTIONALLY-PROTECTED{ }**  
**FROM EnhancedSecurity** enhancedSecurity

**id-errcode-dsaReferral**  
**FROM DirectoryAccessProtocol** dap

*-- parameterized type for deriving chained operations --*

**chained** { **OPERATION** : operation } **OPERATION ::=** {  
**ARGUMENT** **OPTIONALLY-PROTECTED** {  
**SET** {  
 chainedArgument ChainingArguments,  
 argument [0] operation.&ArgumentType },  
 DIRQOP.&dspChainedOp-QOP{@dirqop} }  
**RESULT** **OPTIONALLY-PROTECTED** {  
**SET** {  
 chainedResult ChainingResults,  
 result [0] operation.&ResultType },  
 DIRQOP.&dspChainedOp-QOP{@dirqop} }  
**ERRORS** { operation.&Errors EXCEPT (referral | dsaReferral) }  
**CODE** operation.&operationCode }

*-- bind and unbind operations --*



```

dSABind          OPERATION ::= directoryBind
dSAUnbind        OPERATION ::= directoryUnbind
-- chained operations --
chainedRead      OPERATION ::= chained { read }
chainedCompare   OPERATION ::= chained { compare }
chainedAbandon   OPERATION ::= abandon
chainedList      OPERATION ::= chained { list }
chainedSearch    OPERATION ::= chained { search }
chainedAddEntry  OPERATION ::= chained { addEntry }
chainedRemoveEntry OPERATION ::= chained { removeEntry }
chainedModifyEntry OPERATION ::= chained { modifyEntry }
chainedModifyDN  OPERATION ::= chained { modifyDN }
-- errors and parameters --
dsaReferral ERROR ::= {
    PARAMETER    OPTIONALLY-PROTECTED {
        SET {
            reference      [0] ContinuationReference,
            contextPrefix  [1] DistinguishedName OPTIONAL,
            COMPONENTS OF CommonResults },
        DIRQOP.&dsaReferral-QOP{@dirqop} }
    CODE         id-errcode-dsaReferral }
-- common arguments and results --
ChainingArguments ::= SET {
    originator          [0] DistinguishedName OPTIONAL,
    targetObject        [1] DistinguishedName OPTIONAL,
    operationProgress   [2] OperationProgress
                        DEFAULT { nameResolutionPhase notStarted },
    traceInformation    [3] TraceInformation,
    aliasDereferenced   [4] BOOLEAN DEFAULT FALSE,
    aliasedRDNs         [5] INTEGER OPTIONAL,
                        -- only present in 1988 systems
    returnCrossRefs     [6] BOOLEAN DEFAULT FALSE,
    referenceType       [7] ReferenceType DEFAULT superior,
    info                [8] DomainInfo OPTIONAL,
    timeLimit           [9] UTCTime OPTIONAL,
    securityParameters  [10] SecurityParameters DEFAULT {},
    entryOnly           [11] BOOLEAN DEFAULT FALSE,
    uniqueIdentifier    [12] UniqueIdentifier OPTIONAL,
    authenticationLevel [13] AuthenticationLevel OPTIONAL,
    exclusions          [14] Exclusions OPTIONAL,
    excludeShadows      [15] BOOLEAN DEFAULT FALSE,
    nameResolveOnMaster [16] BOOLEAN DEFAULT FALSE,
    operationIdentifier [17] INTEGER OPTIONAL }

ChainingResults ::= SET {
    info                [0] DomainInfo OPTIONAL,
    crossReferences     [1] SEQUENCE OF CrossReference OPTIONAL,
    securityParameters  [2] SecurityParameters DEFAULT {},
    alreadySearched     [3] Exclusions OPTIONAL }

CrossReference ::= SET {
    contextPrefix [0] DistinguishedName,
    accessPoint  [1] AccessPointInformation }

```

```

ReferenceType ::= ENUMERATED {
    superior          (1),
    subordinate       (2),
    cross             (3),
    nonSpecificSubordinate (4),
    supplier          (5),
    master            (6),
    immediateSuperior (7),
    self              (8) }

```

TraceInformation ::= SEQUENCE OF TraceItem

```

TraceItem ::= SET {
    dsa                [0]Name,
    targetObject       [1]Name OPTIONAL,
    operationProgress  [2]OperationProgress }

```

```

OperationProgress ::= SET {
    nameResolutionPhase [0]    ENUMERATED {
        notStarted (1),
        proceeding (2),
        completed (3) },
    nextRDNTToBeResolved [1]    INTEGER OPTIONAL }

```

DomainInfo ::= ABSTRACT-SYNTAX.&Type

```

ContinuationReference ::= SET {
    targetObject       [0]    Name,
    aliasedRDNs        [1]    INTEGER OPTIONAL, -- only present in 1988 systems
    operationProgress  [2]    OperationProgress,
    rdnsResolved       [3]    INTEGER OPTIONAL,
    referenceType      [4]    ReferenceType,
    accessPoints       [5]    SET OF AccessPointInformation,
    entryOnly          [6]    BOOLEAN DEFAULT FALSE,
    exclusions         [7]    Exclusions OPTIONAL,
    returnToDUA        [8]    BOOLEAN DEFAULT FALSE,
    nameResolveOnMaster [9]    BOOLEAN DEFAULT FALSE }

```

```

AccessPoint ::= SET {
    ae-title           [0]    Name,
    address            [1]    PresentationAddress,
    protocolInformation [2]    SET OF ProtocolInformation OPTIONAL }

```

```

AccessPointInformation ::= SET {
    COMPONENTS OF      MasterOrShadowAccessPoint ,
    additionalPoints   [4]    SET OF MasterOrShadowAccessPoint OPTIONAL }

```

```

MasterOrShadowAccessPoint ::= SET {
    COMPONENTS OF      AccessPoint,
    category           [3]    ENUMERATED {
        master (0),
        shadow (1) } DEFAULT master } }

```

MasterAndShadowAccessPoints ::= SET OF MasterOrShadowAccessPoint

Exclusions ::= SET OF RDNSSequence

END

---

Anexo B

Ejemplo de resolución de nombre distribuida

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

En la figura B.1 se presenta un ejemplo que ilustra la manera de utilizar la resolución de nombre distribuida para procesar diferentes peticiones de directorio. El ejemplo se basa en el DIT hipotético y la configuración o configuraciones de DSA correspondientes descritas en el anexo M (Modelación de conocimiento) de la Rec. UIT-T X.501 | ISO/CEI 9594-2, reproducido aquí para mayor comodidad.

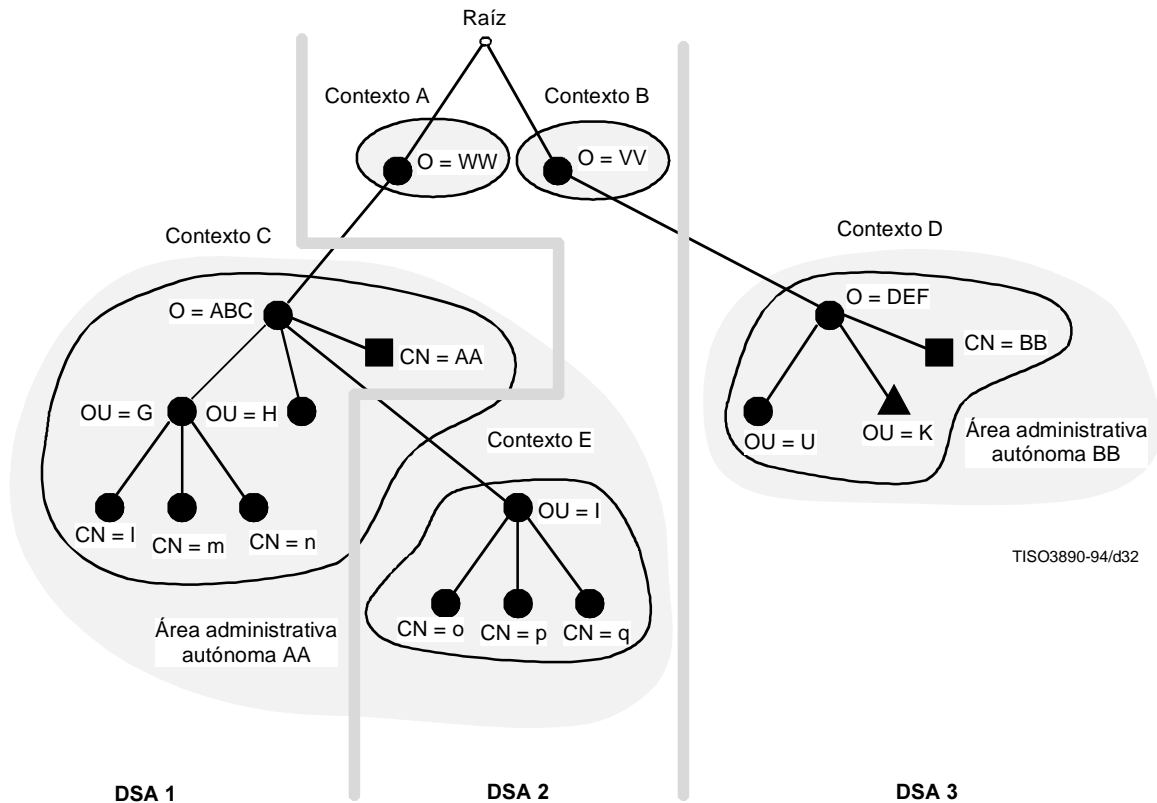


Figura B.1 – DIT hipotético mapeado a tres DSA

Suponiendo un modo de propagación por concatenación, las siguientes peticiones dirigidas a DSA 1 se procesarían de la forma siguiente:

- 1) Una petición con el nombre distinguido {C = WW, O = ABC, OU = G, CN = 1}
  - El procedimiento de resolución de nombre concordará con éxito cada RDN en el nombre de destino con DSE contenidos por DSA 1, hasta localizar la DSE de destino.
- 2) Una petición con el nombre distinguido {C = WW, O = JPR}
  - El procedimiento de resolución de nombre en DSA 1 concordará la DSE C = WW, y será incapaz de continuar la concordancia. En este punto, DSA 1 podrá encontrar dos referencias que le facilitarán la prosecución: una es la referencia **immSupr** en DSE C = WW, y la otra es la referencia **supr** en la DSE raíz. En este ejemplo hipotético, las dos referencias estarían apuntando a DSA 2. Por tanto, la petición es concatenada a DSA 2.
  - En DSA 2, el procedimiento de **resolución de nombre** concordará la DSE C = WW, y será incapaz de continuar la concordancia. En este caso, dado que la DSE C = WW es una **cp** y **entry**, y DSA 2 es el DSA maestro para esta inserción, y que, por añadidura, no hay **nssr** en C = WW, DSA 2 es entonces capaz de determinar que no existe ese nombre en el directorio. Se retorna un error de nombre **noSuchObject**.

- 3) Una petición con nombre distinguido {C = VV, O = DEF, OU = K}
- El procedimiento de **resolución de nombre** en DSA 1 no será capaz de concordar ninguna DSE. La única referencia disponible es la referencia **supr** en la DSE raíz, que apunta a DSA 2. Por tanto, la petición es concatenada a DSA 2.
  - En DSA 2, el procedimiento de **resolución de nombre** concordará la DSE C = VV, y después, DSE O = DEF, y será incapaz de continuar la concordancia. Como se halla que DSE O = DEF es de tipo **subr**, la referencia de conocimiento específica, que apunta a DSA 3, se utiliza esta referencia, y la petición es concatenada a DSA 3.
  - En DSA 3, el procedimiento de **resolución de nombre** concordará el nombre objeto de destino completo, y encontrará que la DSE localizada es de tipo **alias**. Suponiendo que en este caso hay que desreferenciar los alias, se construirá un nuevo nombre utilizando el **aliasedEntryName** contenido en la DSE concordado. El DSA 3 reentrará entonces el procedimiento de **resolución de nombre** para continuar.

## Anexo C

### Uso distribuido de la autenticación

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

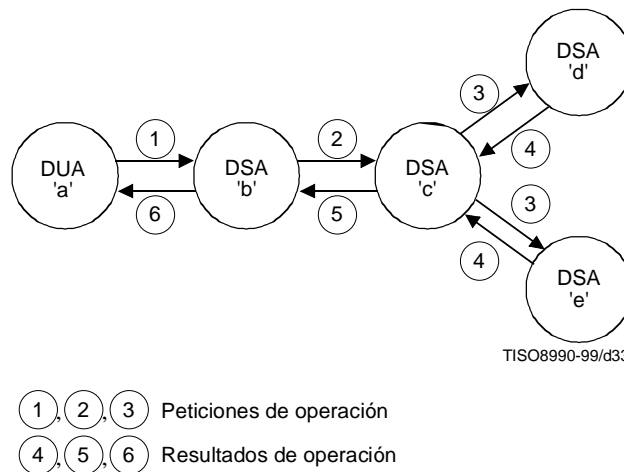
#### C.1 Resumen

El modelo de seguridad está definido en la cláusula 10 de la Rec. UIT-T X.501 | ISO/CEI 9594-2. A continuación se resumen los principales aspectos del modelo:

- a) La autenticación fuerte, mediante la firma de la petición, el resultado y los errores, es soportada en el DSP.
- b) La criptación de la petición, el resultado y los errores es soportada en el DSP.

En este anexo se describe la forma según la cual se realiza lo indicado en el directorio distribuido. Se utiliza la terminología y la notación definidas en la Rec. UIT-T X.509 | ISO/CEI 9594-8.

#### C.2 Modelo de protección distribuida



**Figura C.1 – Modelo de protección distribuida**

En la figura C.1 se ilustra el modelo que debe utilizarse para especificar los procedimientos de protección distribuida. El modelo identifica la secuencia de flujos de información para el caso general de una operación de listar o buscar. Se considera que una operación se origina en el DUA 'a' citando un objeto de destino que reside en el DSA 'c'; en la ejecución de la operación participan los DSA 'b', 'c', 'd' y 'e'.

Inicialmente el DUA 'a' se pone en contacto con cualquier DSA (DSA 'b') que no contiene el objeto de destino pero que puede navegar, mediante concatenación, hacia el DSA que lo contiene (DSA 'c'). Si todos los DSA estuvieran funcionando en modo referimiento, el modelo se simplificaría notablemente y cada intercambio DSA/DSA equivaldría, desde el punto de vista de la protección, a la interacción entre el DUA 'a' y el DSA 'b'.

##### C.2.1 Calidad de la protección

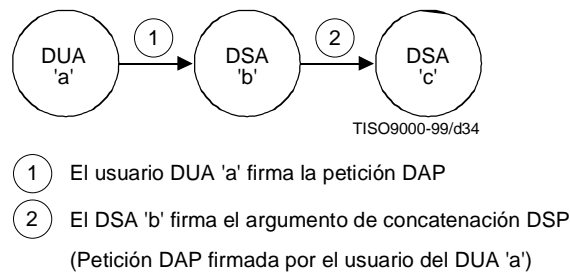
La calidad de la protección que se ha de utilizar durante la vida útil de la asociación de aplicación se establece durante la operación de vinculación de directorio. La política del sistema determinará el nivel de protección que deberán respetar el DUA y el DSA. **DIRQOP** es una clase de objeto de información que puede utilizarse para especificar la calidad de la protección que se ha de asociar a cada operación (petición, resultado o errores). El DUA transmite la clase de objeto de información **DIRQOP** en el **DirectoryBindArgument**, y el DSA acepta este nivel de protección en el **DirectoryBindResult**. La calidad de la protección se puede utilizar para proporcionar los siguientes tipos de protección: firmada, criptada, o firmada y criptada.

### C.3 Operaciones concatenadas firmadas

Si se soportan las operaciones concatenadas firmadas digitalmente, el DUA es responsable de verificar las firmas digitales retornadas por el DSA en un resultado de listar o buscar. El DUA debe ser capaz de verificar las firmas digitales de más de un DSA si se utiliza un entorno distribuido para generar los resultados de listar o buscar. La correlación de los resultados de las operaciones de listar y buscar es responsabilidad del DUA. Los DSA no deberían fusionar estos resultados en nombre del DUA. En ciertos casos, el DUA puede recibir información de varios DSA, cada uno de los cuales soporta diferentes niveles de autenticación y firmas digitales. El DUA debe tomar la decisión de utilizar o no la información devuelta si la firma digital no es válida.

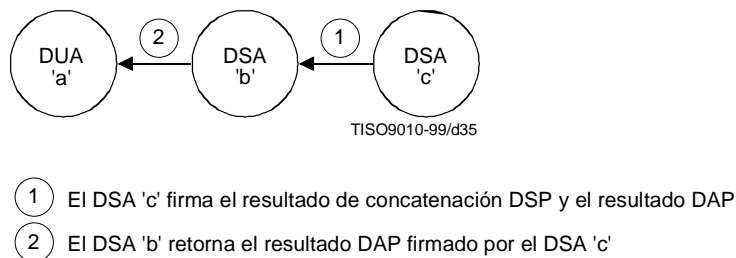
#### C.3.1 Argumentos firmados concatenados

Si un argumento DAP es firmado por el DUA, la firma debe mantenerse mientras exista la petición. Los DSA pueden verificar y utilizar esta firma al efectuar las verificaciones de control de acceso. Si el DSA determina que, para el procesamiento, es necesario concatenar la petición con otro DSA, incluirá la petición firmada del DUA con los necesarios argumentos de concatenación. Si el DSA va a soportar las operaciones DSP firmadas (DSA a DSA), las credenciales del DSA se utilizarían para firmar los **ChainingArguments** DSP y la firma del DUA se mantendría junto con la petición DAP original.



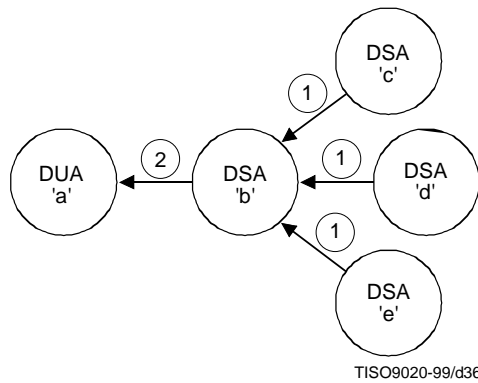
#### C.3.2 Resultados firmados concatenados

Si el usuario DUA desea recibir resultados firmados de directorio, el campo **SecurityParameters.ProtectionRequest** debe fijarse a **SIGNED**. El DSA distante debe poderse configurar como para enviar **ChainingResults** firmados digitalmente. Facultativamente, el DSA distante puede firmar el resultado DAP y los **ChainingResults** DSP, soportando de ese modo las firmas de extremo a extremo. El DSA 'b' será responsable de verificar la firma DSP del DSA distante, y el DUA 'a' será responsable de verificar la firma de resultado DAP del DSA.



#### C.3.3 Fusión de los resultados firmados de listar o buscar

El DUA debe ser capaz de verificar las firmas digitales de más de un DSA si se utiliza un entorno distribuido para generar los resultados de listar o buscar. La correlación de los resultados de las operaciones listar y buscar es responsabilidad del DUA. Los DSA no deberían fusionar estos resultados en nombre del usuario DUA. En ciertos casos, el DUA puede recibir información de varios DSA, cada uno de los cuales soporta niveles diferentes de autenticación y firmas digitales. Si la firma digital no es válida, el DUA debe decidir si utilizará o no la información devuelta.



- ① Los DSA 'c', 'd' y 'e' firman el resultado de concatenación DSP (resultado DAP firmado por los DSA 'c', 'd' y 'e')
- ② El DSA 'b' retorna el resultado DAP parcial firmado por los DSA 'c', 'd' y 'e'; el DSA 'b' no fusiona los resultados DAP

NOTA – El protocolo DSP de DSA a DSA también se puede firmar, criptar, o firmar y criptar.

### C.3.4 Petición de concatenación múltiple

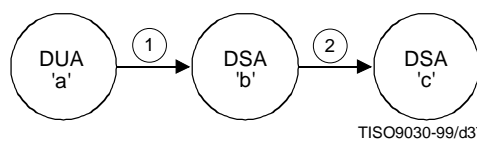
Si el DSA determina que es necesario concatenar la petición DAP a otros múltiples DSA, puede multiconcatenar la petición, ya sea en paralelo o secuencialmente. Se describen dos modos de descomposición: referencias subordinadas no específicas NSSR, *non-specific subordinate reference* o descomposición de petición. En el caso de la descomposición NSSR, el DSA envía la petición idéntica a otros DSA identificados. En el caso de la descomposición de petición, el DSA envía peticiones subsiguientes parciales (posiblemente diferentes) a cada uno de los otros DSA.

## C.4 Operaciones concatenadas criptadas

Si se soporta la criptación, es necesario conferir una protección equivalente entre cada uno de los componentes de directorio. Para llegar a un acuerdo respecto de la equivalencia de políticas se necesitan correspondencias, que están más allá del alcance de esta especificación.

### C.4.1 Criptación de petición punto a punto (DUA->DSA o DSA->DSA)

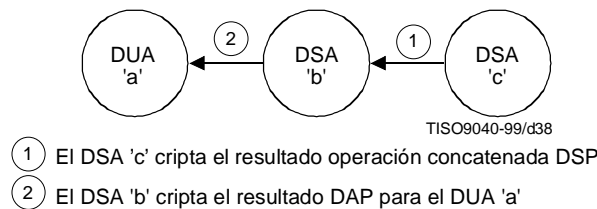
Si un usuario DUA desea criptar la petición DAP, la criptación sólo puede tener lugar sobre una base punto a punto. Aunque el DUA criptará la petición DAP para el DSA 'b', el usuario DUA no sabe si la petición se concatenará en última instancia a un DSA distante para su procesamiento. El DSA 'b' descryptará la petición y tratará de satisfacerla. Si el DSA 'b' determina que la petición debería concatenarse a otro DSA (DSA 'c') para el procesamiento, el DSA 'b' debe criptar las operaciones concatenadas para el DSA 'c'. La selección de protección punto a punto para la petición y las respuestas DSP (**argumentos y resultados de operación concatenada**) se indica mediante el **dirqop** establecido entre el DSA 'b' y el DSA 'c' en la vinculación DSP.



- ① El usuario DUA 'a' cripta la petición DAP para el DSA 'b'
- ② El DSA 'b' cripta el argumento operación concatenada DSP

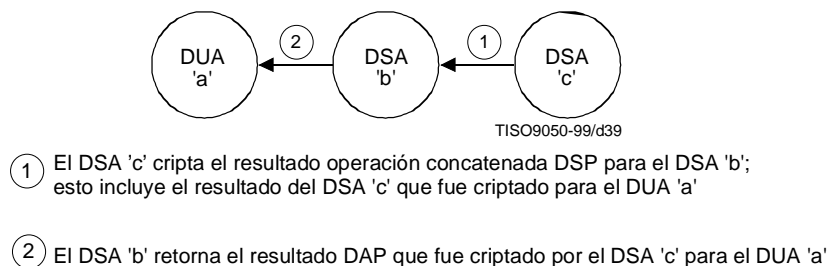
### C.4.2 Criptación de resultado punto a punto (DUA<-DSA o DSA<-DSA)

Si el usuario DUA desea recibir resultados o errores criptados de directorio, el campo **SecurityParameters.ProtectionRequest** debe fijarse a **ENCRYPTED** o, si este campo no está presente, el campo **SecurityParameters.ProtectionRequest** en los **chained operation Arguments** se debe fijar como para reflejar el **DIRQOP** en el **BindArgument** DAP. El DSA distante (DSA 'c') debe poder configurarse como para enviar **chained operation Results** criptados. Conforme a esta hipótesis, el sistema DSA 'c' determina si puede satisfacer la petición, y genera un resultado DAP y **chained operation Results** DSP. El DSA 'c' puede efectuar la criptación punto a punto mediante la criptación de **chained operation Results** DSP para el DSA 'b'. El DSA 'b' puede descryptar los resultados de la operación concatenada DSP y criptar el resultado DAP para el usuario DUA 'a'. Esto conduce a la criptación punto a punto del resultado. El DUA 'a' será responsable de descryptar el resultado DAP del DSA local (DSA 'b').



#### C.4.3 Criptación de extremo a extremo del resultado DAP y criptación de punto a punto del resultado de concatenación DSP

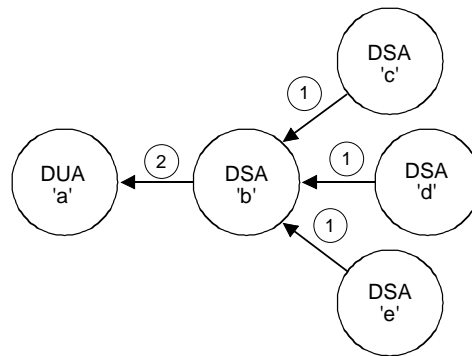
Si el usuario del DUA 'a' desea recibir resultados o errores criptados de directorio, el campo **SecurityParameters.ProtectionRequest** debe fijarse a **ENCRYPTED** o, si este campo no está presente, el campo **SecurityParameters.ProtectionRequest** en los argumentos de operación concatenada se debe fijar como para reflejar el **DIRQOP** en la vinculación DAP. El DSA 'c' distante debe poder configurarse como para enviar resultados de operación concatenada criptados. Conforme a esta hipótesis, el sistema DSA 'c' determina que puede satisfacer la petición, genera una criptación de extremo a extremo en el resultado DAP (para el usuario DUA) y una criptación de punto a punto en el resultado de operación concatenada DSP. El DSA 'c' puede efectuar la criptación de extremo a extremo porque sabe quién es el usuario del DUA 'a' previsto. El DSA 'c' puede efectuar la criptación de punto a punto en los resultados de operación concatenada DSP, mediante la criptación de los resultados de operación concatenada DSP para el DSA1. El DSA 'b' puede descriptar el DSP y retransmitir el resultado DAP criptado al usuario del DUA 'a'. El DUA será responsable de descriptar el resultado DAP que reciba del DSA 'c' a través del DSA 'b'.



#### C.4.4 Fusión de los resultados de listar/buscar (fusión con recriptación por el DSA1)

Si el usuario del DUA 'a' desea recibir el directorio los resultados o errores criptados de listar o buscar, el campo **SecurityParameters.ProtectionRequest** debe fijarse a **ENCRYPTED** o, si este campo no está presente, el campo **SecurityParameters.ProtectionRequest** en los argumentos de operación concatenada se debe fijar como para reflejar el **DIRQOP** en la vinculación DAP. El DSA local (DSA 'b') puede decidir efectuar una concatenación múltiple de la petición listar/buscar con varios otros DSA (ya sean en paralelo o secuencialmente). Los DSA distantes (DSA 'c', 'd' y 'e') deben poder configurarse como para enviar resultados concatenados de listar/buscar criptados. Conforme a este modelo, cada uno de los DSA distantes ('c', 'd' y 'e') satisface la petición y genera resultados DAP y resultados de operación concatenada DSP criptados. Los resultados de operación concatenada generados por los DSA distantes ('c', 'd' y 'e') se transfieren al DSA 'b'. El DSA 'b' recibe cada uno de los resultados de operación concatenada, descripta los resultados y coteja o fusiona los resultados en un resultado común. Luego el DSA 'b' cripta este nuevo resultado común de listar/buscar y lo envía al usuario del DUA 'a'. Los DSA distantes efectúan la criptación punto a punto mediante la criptación de los resultados de operación concatenada DSP para el DSA 'b' y mediante la criptación, por el DSA 'b', del resultado DAP para el usuario del DUA 'a'. El DUA será responsable de descriptar un resultado DAP fusionado.





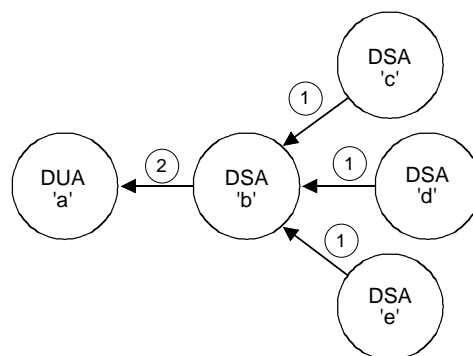
TISO9060-99/d40

- ① Los DSA 'c', 'd' y 'e' criptan los resultados de la operación concatenada DSP (incluido el resultado DAP)
- ② El DSA 'b' describe los resultados de la operación concatenada DSP de los DSA 'c' y 'd', y luego el DSA fusiona los resultados DAP y vuelve a criptar el resultado DAP del DUA 'a'

#### C.4.5 Fusión no autorizada para los resultados de listar/buscar

**(No fusión por el DSA 'b' proporcionando criptación de extremo a extremo del resultado DAP de listar/buscar)**

Si el usuario DUA desea recibir el directorio los resultados o errores criptados de listar o buscar, el campo **SecurityParameters.ProtectionRequest** debe ponerse a **ENCRYPTED** o, si este campo no está presente, el campo **SecurityParameters.ProtectionRequest** en los argumentos de operación concatenada se fija como para reflejar el **DIRQOP** en la vinculación DAP. El DSA local puede decidir efectuar la concatenación múltiple de la petición de listar/buscar con varios otros DSA (ya sea en paralelo o secuencialmente). Los DSA distantes ('c', 'd' y 'e') deben poder configurarse como para enviar resultados concatenadas de listar/buscar criptados. Según esta hipótesis, cada uno de los DSA distantes ('c', 'd' y 'e') satisface la petición y genera resultados DAP criptados (para el usuario del DUA 'a') y resultados de operación concatenada DSP criptados (para el DSA 'b'). Los resultados de operación concatenada generados por los DSA distantes ('c', 'd' y 'e') se transfieren al DSA 'b'. Este DSA recibe cada uno de los resultados de operación concatenada, describe los resultados de operación concatenada DSP y NO efectúa ningún tipo de comparación o fusión de los resultados. El DSA 'b' retransmite los resultados de listar/buscar (que fueron criptados por los DSA 'c', 'd' y 'e') y los envía al DUA 'a' sin modificaciones. Los DSA distantes efectúan la criptación de extremo a extremo mediante la criptación del resultado DAP de listar/buscar para el usuario del DUA 'a' y el DSA distante efectúa la criptación de punto a punto mediante la criptación de los resultados de operación concatenada DSP para el DSA 'b'. El DUA 'a' será responsable de descriptar cada uno de los resultados DAP de listar/buscar devueltos.

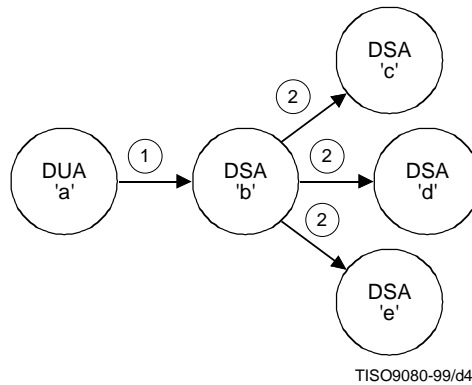


TISO9070-99/d41

- ① Los DSA 'c', 'd' y 'e' criptan los resultados de la operación concatenada DSP para el DSA 'b'; esto incluye los resultados que han sido criptados para el usuario del DUA 'a'
- ② El DSA 'b' describe los resultados de la operación concatenada DSP 'c', y 'd', y luego el DSA transmite los resultados DAP (que fueron criptados por cada uno de los DSA 'c', 'd' y 'e' para el DUA 'a') al DUA 'a', sin descriptarlos ni fusionarlos

### C.4.6 Concatenación múltiple de una petición DAP utilizando una clave-criptación (clave-red)

Si el usuario del DUA 'a' desea recibir resultados o errores criptados de directorio, el campo **SecurityParameters.ProtectionRequest** debe fijarse a **ENCRYPTED** o, si este campo no está presente, el campo **SecurityParameters.ProtectionRequest** en los argumentos de operación concatenada se fija como para reflejar el **DIRQOP** en la vinculación DAP. El DSA local puede decidir efectuar la concatenación múltiple de la petición listar/buscar con varios otros DSA (ya sea en paralelo o secuencialmente). El DSA local (DSA 'b') puede estar configurado como para soportar una clave-criptación o clave-red. Una clave-red es una clave de criptación simétrica que comparten todos los DSA en la cadena. Gracias a la utilización de una clave-red, el DSA 'b' únicamente debe criptar una sola vez la petición concatenada. Cada uno de los DSA distantes están en conocimiento de la clave-red y es capaz de descriptar el argumento de operación concatenada DSP utilizando esa clave-red. Según esta hipótesis, el usuario del DUA puede efectuar la criptación de punto a punto mediante la criptación de la petición DAP para el DSA 'b', y este DSA puede efectuar la criptación punto a punto para los DSA distantes utilizando una clave-red.

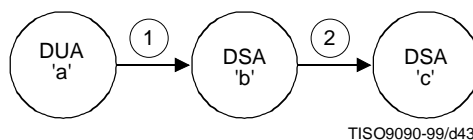


- ① El DUA 'a' cripta un argumento DAP para el DSA 'b'
- ② El DSA 'b' descripta la petición y trata de satisfacerla; si el DSA 'b' no puede satisfacer la petición, utiliza una "clave-red" para criptar la petición operación concatenada DSA (incluida la petición DAP). La petición concatenada se envía a los DSA 'c', 'd' y 'e'

## C.5 Operaciones distribuidas firmadas y criptadas

### C.5.1 Firmas de extremo a extremo, con criptación punto a punto

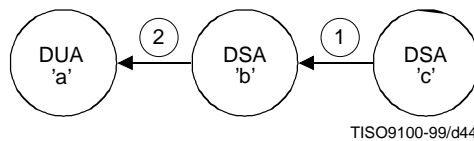
Si un usuario del DUA 'a' desea firmar y criptar la petición DAP, la firma puede proporcionarse de extremo a extremo y la criptación sólo puede tener lugar de punto a punto. El DUA 'a' puede firmar y criptar la petición DAP para el DSA 'b'; no obstante, el usuario del DUA 'a' no sabe si la petición se concatenará en última instancia a un DSA distante (DSA 'c') para el procesamiento. El DAP 'b' descriptará la petición y verificará la firma. Luego tratará de satisfacer la petición. Si el DSA 'b' determina que la petición debe contenerse con otro DSA (DSA 'c') para el procesamiento, el DSA 'b' debe criptar los **ChainingArguments** DSP para el DSA 'c'. La petición DAP firmada original puede mantenerse y transmitirse con los **ChainingArguments** DSP criptados.



- ① El usuario DUA firma y cripta la petición DAP para el DSA 'b'
- ② El DSA 'b' descripta la petición DAP y verifica la firma, después de tratar de satisfacer la petición localmente; el DSA 'b' determina que es preciso concatenar esta petición con el DSA 'c'. El DSA 'b' envía la petición DAP firmada originalmente (firmada por el usuario del DUA 'a') y genera y cripta el argumento de concatenación DSP para el DSA 'c'

**C.5.2 Firma y criptación de extremo a extremo en el resultado DAP, firma y criptación de punto a punto en el DSP**

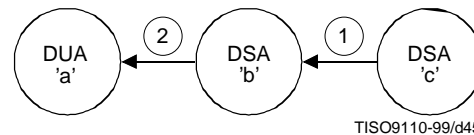
Si el usuario DUA 'a' desea recibir resultados firmados y criptados de directorio, el campo **SecurityParameters.ProtectionRequest** debe ponerse a **SIGNED-AND-ENCRYPTED** o, si este campo no está presente, el campo **SecurityParameters.ProtectionRequest** en los **ChainingArguments** se fija como para reflejar el **DIRQOP** en la vinculación DAP. El DSA distante debe poder configurarse como para enviar operaciones concatenadas firmadas y criptadas. Conforme a este modelo, el sistema DSA 'c' puede satisfacer la petición y genera y efectúa la criptación de extremo a extremo en el resultado DAP (para el usuario del DUA 'a') y una criptación de punto a punto en el DSP. El DSA 'c' puede efectuar la firma y la criptación de extremo a extremo porque sabe quién es el usuario DUA 'a' previsto. El DSA 'c' puede efectuar la firma y criptación de punto a punto en los **ChainingResults** DSP mediante la firma y la criptación de los **ChainingResults** DSP para el DSA 'b'. El DSA 'b' puede descriptar y verificar la firma del DSA 'c' para los **ChainingResults** DSP firmados y retransmitir el resultado DAP firmado y criptado al usuario del DUA 'a'. El DUA 'a' será responsable de descriptar y verificar la firma del resultado DAP que recibe del DSA 'c' a través del DSA 'b'.



- ① El DSA 'c' firma y cripta el resultado concatenado DSP para el DSA 'b'; esto incluye los resultados DAP que son firmados y criptados para el usuario del DUA 'a'
- ② El DSA 'b' descripta el resultado concatenado DSP procedente del DSA 'c' y transmite el resultado DAP firmado y criptado para el DUA 'a'

**C.5.3 Firma de extremo a extremo en el resultado DAP, criptación de punto a punto en el DSP y el resultado DAP**

Si el usuario DUA 'a' desea recibir resultados firmados y criptados de directorio, el campo **SecurityParameters.ProtectionRequest** debe ponerse a **SIGNED-AND-ENCRYPTED** o, si este campo no está presente, el campo **SecurityParameters.ProtectionRequest** en los **ChainingArguments** se fija como para reflejar el **DIRQOP** en la vinculación DAP. El DSA distante (DSA 'c') debe poder configurarse como para enviar operaciones concatenadas firmadas y criptadas. Con arreglo a este modelo, el sistema DSA 'c' puede satisfacer la petición, genera un resultado DAP firmado y firma y cripta el resultado DAP y los **ChainingResults** DSP para el DSA 'b'. El DSA 'b' puede descriptar y verificar la firma del DSA 'c' en los **ChainingResults** DSP y volver a criptar el resultado DAP firmado (por el DSA 'c') para el usuario del DUA 'a'. El DUA 'a' será responsable de descriptar el resultado DAP recibido del DSA 'b' y verificar la firma del resultado DAP que recibe del DSA 'c' a través del DSA 'b'.



- ① El DSA 'c' firma y cripta el resultado concatenado DSP para el DSA 'b'; esto incluye los resultados DAP
- ② El DSA 'b' descripta el resultado concatenado DSP procedente del DSA 'c' (y el resultado DAP recibido en el resultado concatenado DSP) y transmite el resultado DAP firmado al DUA 'a'

## Anexo D

Especificación de tipos de vinculaciones operacionales jerárquicas  
y jerárquicas no específicas

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

Este anexo incluye las definiciones de las clases de objeto de información ASN.1 introducidas en esta Especificación de directorio en forma del módulo ASN.1 **HierarchicalOperationalBindings**.

**HierarchicalOperationalBindings****{joint-iso-itu-t ds(5) module(1) hierarchicalOperationalBindings(20) 3}****DEFINITIONS ::=****BEGIN****-- EXPORTS All --**

*-- The types and values defined in this module are exported for use in the other ASN.1 modules contained  
-- within the Directory Specifications, and for the use of other applications which will use them to access  
-- Directory services. Other applications may use them for their own purposes, but this will not constrain  
-- extensions and modifications needed to maintain or improve the Directory service.*

**IMPORTS**

**informationFramework, distributedOperations, directoryOperationalBindingTypes,  
opBindingManagement, dsp**

**FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 3}**

**Attribute, RelativeDistinguishedName, DistinguishedName**

**FROM InformationFramework informationFramework**

**MasterAndShadowAccessPoints**

**FROM DistributedOperations distributedOperations**

**directorySystemAC**

**FROM DirectorySystemProtocol dsp**

**OPERATIONAL-BINDING**

**FROM OperationalBindingManagement opBindingManagement**

**id-op-binding-hierarchical, id-op-binding-non-specific-hierarchical**

**FROM DirectoryOperationalBindingTypes directoryOperationalBindingTypes ;**

*-- types --*

**HierarchicalAgreement ::= SEQUENCE {**  
**rdn [0] RelativeDistinguishedName,**  
**immediateSuperior [1] DistinguishedName }**

**NonSpecificHierarchicalAgreement ::= SEQUENCE {**  
**immediateSuperior [1] DistinguishedName }**

**SuperiorToSubordinate ::= SEQUENCE {**  
**contextPrefixInfo [0] DITcontext,**  
**entryInfo [1] SET OF Attribute OPTIONAL,**  
**immediateSuperiorInfo [2] SET OF Attribute OPTIONAL }**

**DITcontext ::= SEQUENCE OF Vertex**

```
Vertex ::= SEQUENCE {
    rdn [0] RelativeDistinguishedName,
    admPointInfo [1] SET OF Attribute OPTIONAL,
    subentries [2] SET OF SubentryInfo OPTIONAL,
    accessPoints [3] MasterAndShadowAccessPoints OPTIONAL }
```

```
SubentryInfo ::= SEQUENCE {
    rdn [0] RelativeDistinguishedName,
    info [1] SET OF Attribute }
```

```
SubordinateToSuperior ::= SEQUENCE {
    accessPoints [0] MasterAndShadowAccessPoints OPTIONAL,
    alias [1] BOOLEAN DEFAULT FALSE,
    entryInfo [2] SET OF Attribute OPTIONAL,
    subentries [3] SET OF SubentryInfo OPTIONAL }
```

```
SuperiorToSubordinateModification ::= SuperiorToSubordinate (
    WITH COMPONENTS { ..., entryInfo ABSENT})
```

```
NHOBSuperiorToSubordinate ::= SuperiorToSubordinate (
    WITH COMPONENTS { ..., entryInfo ABSENT})
```

```
NHOBSubordinateToSuperior ::= SubordinateToSuperior (
    WITH COMPONENTS { ..., alias ABSENT, entryInfo ABSENT})
```

-- operational binding information objects --

```
hierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
    AGREEMENT HierarchicalAgreement
    APPLICATION CONTEXTS {
        {directorySystemAC} }
    ASYMMETRIC
    ROLE-A { -- superior DSA
        ESTABLISHMENT-INITIATOR TRUE
        ESTABLISHMENT-PARAMETER SuperiorToSubordinate
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER SuperiorToSubordinateModification
        TERMINATION-INITIATOR TRUE }
    ROLE-B { -- subordinate DSA
        ESTABLISHMENT-INITIATOR TRUE
        ESTABLISHMENT-PARAMETER SubordinateToSuperior
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER SubordinateToSuperior
        TERMINATION-INITIATOR TRUE }
    ID id-op-binding-hierarchical }
```

```
nonSpecificHierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
    AGREEMENT NonSpecificHierarchicalAgreement
    APPLICATION CONTEXTS {
        { directorySystemAC } }
    ASYMMETRIC
    ROLE-A { -- superior DSA
        ESTABLISHMENT-PARAMETER NHOBSuperiorToSubordinate
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER NHOBSuperiorToSubordinate
        TERMINATION-INITIATOR TRUE }
    ROLE-B { -- subordinate DSA
        ESTABLISHMENT-INITIATOR TRUE
        ESTABLISHMENT-PARAMETER NHOBSubordinateToSuperior
        MODIFICATION-INITIATOR TRUE
        MODIFICATION-PARAMETER NHOBSubordinateToSuperior
        TERMINATION-INITIATOR TRUE }
    ID id-op-binding-non-specific-hierarchical }
```

END

---

## Anexo E

## Ejemplo de mantenimiento de conocimiento

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

Este anexo ilustra, mediante un ejemplo simple, el mantenimiento de conocimiento, tal como se define en la cláusula 23. En la figura E.1, los símbolos siguientes se utilizan para representar árboles de información de DSA, de cinco DSA.

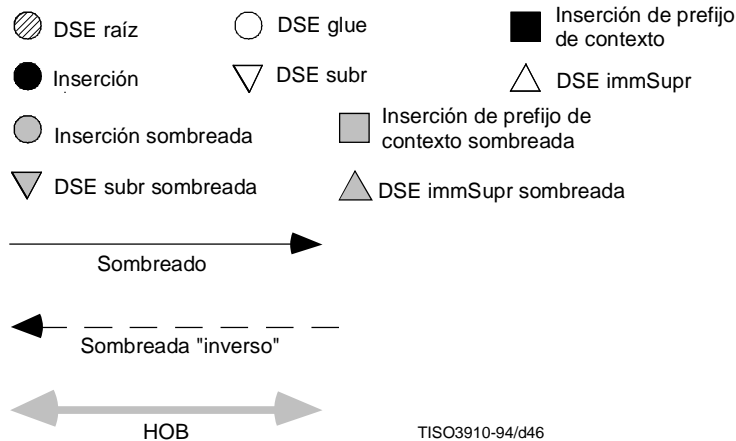


Figura E.1 – Símbolos utilizados para representar árboles de información de DSA

En la figura E.2, DSA 1 es el maestro para el contexto de denominación {A}, constituido por las dos inserciones {A} y {A, B}. DSA 1 contiene una referencia subordinada para contextos de denominación {A, B, C} que es mantenida vía una HOB con DSA 3. DSA 1 es un suministrador de sombra a DSA 2, al que suministra copias de la información de usuario del contexto de denominación {A} y la referencia subordinada a contexto de denominación {A, B, C} que identifica los puntos de acceso de DSA 3, DSA 4 y DSA 5, siendo el primero de ellos el maestro para el contexto de denominación subordinado.

DSA 3 es el maestro para el contexto de denominación {A, B, C}. Además de contener la inserción única {A, B, C} del contexto de denominación, DSA 3 contiene una referencia superior inmediata para el contexto de denominación {A} que es mantenido vía una HOB con DSA 1. DSA 3 es un suministrador de sombra a DSA 4, al que suministra copias de la información de usuario del contexto de denominación {A, B, C} y la referencia superior inmediata al contexto de denominación {A} que identifica los puntos de acceso de DSA 1 y DSA 2, siendo el primero de ellos el maestro para el contexto de denominación superior. DSA 4 es un suministrador de sombra (secundario) a DSA 5, al que proporciona una copia de la información que recibe de DSA 3.

La figura E.2 ilustra los atributos operacionales de DSA empleados para representar y mantener conocimiento.

DSA 1 utiliza el valor de su atributo **myAccessPoint** (asociado con su DSE raíz) y los valores utilizables en común de su atributo **consumerKnowledge** (asociado con el prefijo de contexto {A}) para formar un valor del tipo **MasterAndShadowAccessPoints** para uso en sus interacciones HOB con DSA 3. DSA 3, a su vez, utiliza el valor de su atributo **myAccessPoint** (asociado con su DSE raíz) y los valores utilizables en común de su atributo **consumerKnowledge** y de su atributo **secondaryShadows** (ambos asociados con prefijo de contexto {A, B, C}) para formar un valor del tipo **MasterAndShadowAccessPoints** para uso en sus interacciones HOB con DSA 1. Juntos, los dos DSA, y mediante el DOP, mantienen una referencia subordinada contenida por DSA 1 y una referencia superior inmediata contenida por DSA 3. La referencia subordinada de DSA 1, expresada por un atributo **specificKnowledge** asociado con una DSE en {A, B, C}, se basa en el valor **MasterAndShadowAccessPoints** que recibe del DSA 3; la referencia superior inmediata de DSA 3, expresada por un atributo **specificKnowledge** asociado con una DSE en {A}, se basa similarmente en el valor **MasterAndShadowAccessPoints** que recibe de DSA 1.

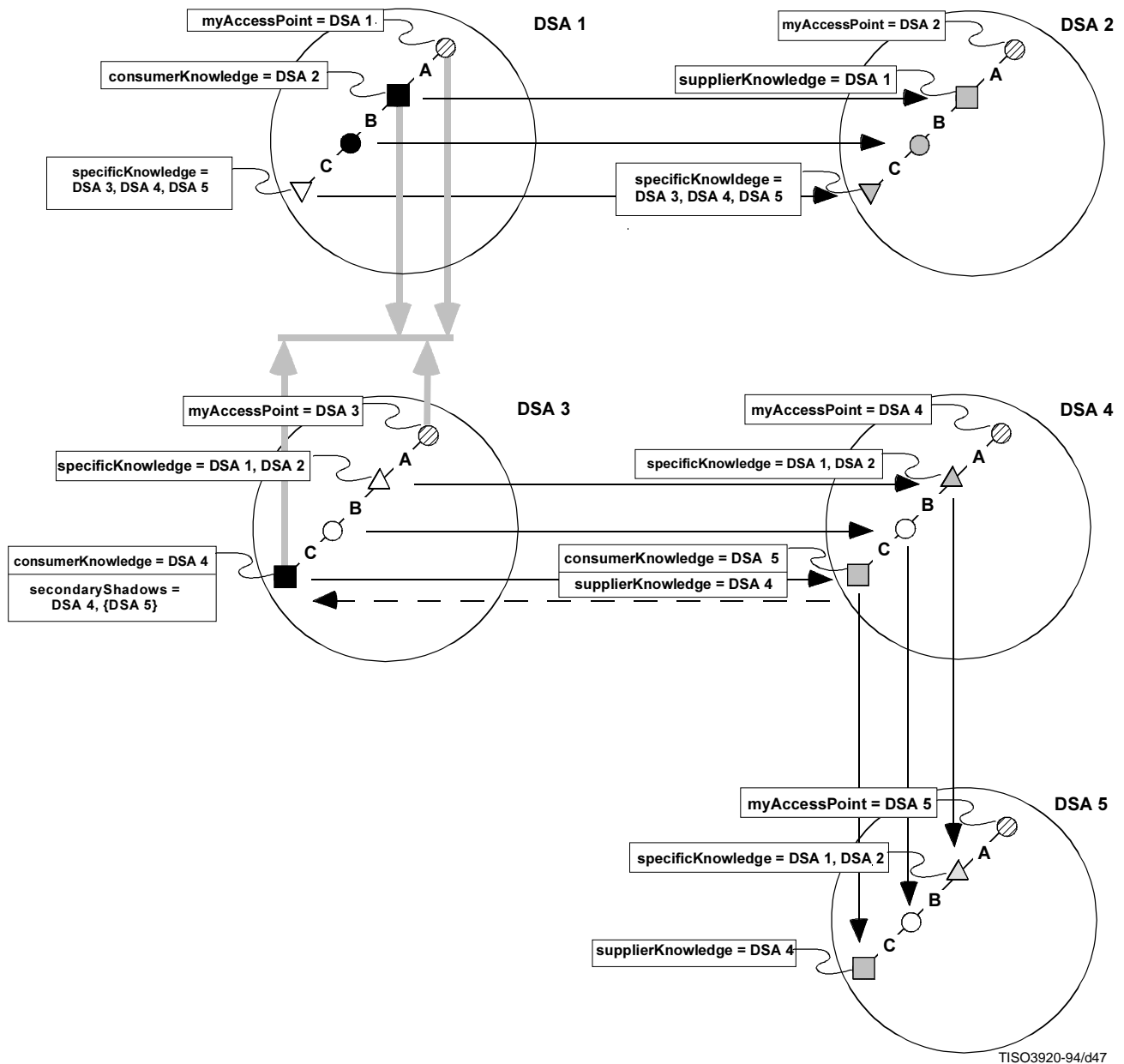


Figura E.2 – Ejemplo de mantenimiento de conocimiento

DSA 1 y DSA 2 utilizan sus valores de **myAccessPoints** en interacciones de vinculaciones operacionales de sombreado para mantener un valor de **consumerKnowledge** en DSA 1 (que identifica el punto de acceso de DSA 2) y **supplierKnowledge** en DSA 2 (que identifica el punto de acceso de DSA 1), estando ambos atributos asociados con el prefijo de contexto {A}. Juntos, los dos DSA, y mediante DOP, mantienen la referencia de consumidor contenida por DSA 1 y la referencia de suministrador contenida por DSA 2.

DSA 2 recibe una copia del atributo **specificKnowledge** asociado con el prefijo de contexto {A, B, C} de DSA 1 en interacción DISP con DSA 1. Esta interacción sirve para mantener la referencia subordinada de DSA 2 al prefijo de contexto {A, B, C}.

DSA 3 y DSA 4 (y de modo similar DSA 4 y DSA 5) mantienen referencias de consumidor y suministrador, respectivamente, de una manera análoga a la interacción entre DSA 1 y DSA 2.

DSA 4 recibe una copia del atributo **specificKnowledge** asociado con el prefijo de contexto {A4} de DSA 3 en interacciones DISP con DSA 3. Esta interacción sirve para mantener la referencia superior inmediata de DSA 4 al prefijo de contexto {A}.

DSA 4 comunica a DSA 3 todo cambio en sus atributos **myAccessPoint** y **consumerKnowledge** (así como en el atributo **secondaryShadows**, que es nulo en este ejemplo), utilizando la operación de vinculación operacional modificar del DOP. DSA 4 suministra a DSA 3 un valor de **SupplierAndConsumers** que contiene solamente aquellos valores del atributo **consumerKnowledge** que identifican los puntos de acceso de DSA que tienen sombras utilizables en común; los valores del atributo **secondaryShadows** suministrados por DSA 4, si es que había alguno, deberán ser, todos ellos, por diseño, utilizables en común. (En este ejemplo se ha supuesto que DSA 5 contiene una copia utilizable en común del contexto de denominación en {A, B, C}). DSA 3 utiliza esta información para mantener un valor de su atributo **secondaryShadows** asociado con el prefijo de contexto {A, B, C}. Este atributo, tal como se ha descrito antes, se utiliza en interacciones DOP con DSA 1 para mantener la referencia subordinada de DSA 1 al prefijo de contexto {A, B, C}.

DSA 5 mantiene su referencia superior inmediata al prefijo de contexto {A} utilizando interacciones DISP con DSA 4 de una manera análoga a las interacciones entre DSA 3 y DSA 4.



## Anexo F

### Enmiendas y correcciones

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

Esta edición de la presente Especificación de directorio incluye las siguientes enmiendas:

- Enmienda 1 para utilización de gestión de sistemas para la administración de directorio.
- Enmienda 2 para ligeras ampliaciones para soportar los requisitos de usuario.
- Enmienda 3 para mejora de la seguridad operacional de directorio.
- Enmienda 4 para contextos.

Esta edición de esta Especificación de directorio abarca los siguientes corrigenda técnicos, que subsanan defectos indicados en los siguientes informes de defectos (algunas partes de algunos de los siguientes corrigenda técnicos pueden haber sido incluidos en las enmiendas que formaron esta edición de esta Especificación de directorio):

- Corrigendo técnico 1 (correspondiente a los informes de defectos 094, 108, 109, 111, 112, 113, 114, 115).
- Corrigendo técnico 2 (correspondiente a los informes de defectos 116, 117, 118, 119, 120, 121, 130, 152, 153, 154, 155, 156, 158, 160, 161, 165, 167).

## SERIES DE RECOMENDACIONES DEL UIT-T

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