UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T X.744.1

(03/2003)

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

SERIE X: REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

Gestión de interconexión de sistemas abiertos – Funciones de gestión y funciones de arquitectura de gestión distribuida abierta

Servicio de gestión de soporte lógico de la RGT mediante arquitectura de intermediario de petición de objetos común

Recomendación UIT-T X.744.1

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	**X.730–X.799**
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999

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

Recomendación UIT-T X.744.1

Servicio de gestión de soporte lógico de la RGT mediante arquitectura de intermediario de petición de objetos común

Resumen

Esta Recomendación es parte de una serie de Recomendaciones que especifican los requisitos de la interfaz de CORBA para comunicación entre un sistema de operaciones (OS) y un elemento de red (NE), entre un sistema de operaciones y un dispositivo de mediación (MD), entre un sistema de operaciones y un adaptador Q (QA) y entre sistemas de operaciones en una red de gestión de las telecomunicaciones (RGT). En esta Recomendación se propone la implementación de un servicio de gestión de soporte lógico basado en CORBA de acuerdo con la Rec. UIT-T X.744. La finalidad de este documento es definir un modelo de CORBA/lenguaje de definición de interfaz similar al definido en la Rec. UIT-T X.744 que utiliza el elemento de servicio de información de gestión común (CMISE). En esta Recomendación, se definen interfaces de granularidad fina y de granularidad gruesa (es decir, fachada) para las funciones de gestión de soporte lógico. Esta Recomendación es conforme a las normas de modelado CORBA de las Recomendaciones UIT-T X.780, X.780.1, Q.816, Q.816.1 y M.3120.

Orígenes

La Recomendación UIT-T X.744.1, preparada por la Comisión de Estudio 4 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la AMNT el 29 de marzo de 2003.

PREFACIO

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

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

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

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

NOTA

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

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

PROPIEDAD INTELECTUAL

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

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

ÍNDICE

Recomendación UIT-T X.744.1

Servicio de gestión de soporte lógico de la RGT mediante arquitectura de intermediario de petición de objetos común

1 Alcance

La presente Recomendación forma parte de una serie de Recomendaciones que especifican los requisitos de la interfaz CORBA para comunicación entre sistemas de operaciones (OS, *operations system*) y un elemento de red (NE, *network element*), entre un sistema de operaciones y un dispositivo de mediación (MD, *mediation device*), entre un sistema de operaciones y un adaptador Q (QA, *Q adapter*) y entre sistemas de operaciones en una red de gestión de las telecomunicaciones (RGT) [1].

La función de gestión de soporte lógico incluye la gestión de un sistema para entrega de soporte lógico y también la gestión de soporte lógico dentro de un sistema.

Hay dos aspectos de soporte lógico que tienen que ser considerados separadamente, y que pueden ser descritos como la visión "latente" (dormant) y la visión "activa" de soporte lógico.

La visión latente de soporte lógico se relaciona con los datos que están almacenados en un sistema gestionado y la manera en la cual son entregados e instalados. En general, los datos son información almacenada, tales como ficheros y tablas de datos, pero pueden ser también ficheros que contienen códigos ejecutables. El alcance de la presente Recomendación abarca la visión latente de soporte lógico.

La visión activa de soporte lógico se relaciona con la gestión de recursos que utilizan el soporte lógico. No hay diferencia real entre esta visión y la visión normal de gestión de recursos. El alcance de la presente Recomendación no incluye la visión activa de soporte lógico. Sin embargo, la relación entre objetos gestionados que representan recursos que utilizan el soporte lógico, y los objetos gestionados que representan el soporte lógico que están utilizando (es decir, la visión latente de soporte lógico) está dentro del alcance de la presente Recomendación.

La presente Recomendación abarca:

- el inicio de la transferencia de soporte lógico;
- el control de soporte lógico posterior a la transferencia;
- la activación de soporte lógico (incluida la actualización y corrección (parches) de la versión);
- la desactivación de soporte lógico;
- el cambio restitutivo de soporte lógico;
- la validación de soporte lógico;
- la indagación de soporte lógico;
- la copia de seguridad de soporte lógico;
- el restablecimiento de soporte lógico.

La presente Recomendación no incluye:

- el mecanismo de transferencia de soporte lógico;
- el almacenamiento físico de soporte lógico (la correspondencia de soporte lógico a un almacenamiento físico de ficheros, por ejemplo, un disco flexible, un disco duro, etc.);
- el formatado de soporte lógico;
- la denominación de productos de soporte lógico;

– la puesta en secuencia de instrucciones de gestión de soporte lógico;
– la supervisión de soporte lógico;
– la gestión de procesos que funcionan en un sistema.

En esta Recomendación, se definen las interfaces de granularidad fina y de granularidad gruesa (es decir, fachada) para las funciones de gestión de soporte lógico. Las interfaces de granularidad fina y de fachada proporcionan el mismo soporte para la gestión de soporte lógico. Ambas pueden ser utilizadas con objetos gestionados de granularidad fina y de granularidad gruesa (por ejemplo, los definidos en la Rec. UIT-T M.3120 [2]).

Las actuales redes de telecomunicaciones contienen un número grande y creciente de sistemas de operaciones y de elementos de red suministrados por vendedores diferentes. El número y la variedad de redes y servicios ha aumentado, creando así una diversidad de necesidades de gestión. Este crecimiento ha resultado en la proliferación de interfaces de comunicación únicos entre sistemas de operaciones y elementos de red. La industria de telecomunicaciones trata de aprovechar la normalización de estas interfaces, diseñadas para lograr la interoperabilidad entre una amplia de gama de servicios de operaciones y elementos de red/adaptadores Q, utilizando dispositivos de mediación cuando proceda, y entre servicios de operaciones.

La finalidad primaria de la presente Recomendación es proporcionar un conjunto de mensajes de aplicación y objetos de soporte asociados para soportar la comunicación a través de interfaces CORBA. Debido a la conveniencia de proporcionar soluciones comunes para la red de gestión de las telecomunicaciones, se prevé que estos mensajes y objetos de soporte sean aplicables a otras interfaces de la RGT o relacionadas con la RGT.

2 Referencias

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

[1] Recomendación UIT-T M.3010 (2000), *Principios para una red de gestión de las telecomunicaciones*.

[2] Recomendación UIT-T M.3120 (2001), *Modelo genérico de información a nivel de red y de elemento de red basado en arquitectura de intermediario de petición de objeto común*.

[3] Recomendación UIT-T Q.816 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común*.

[4] Recomendación UIT-T Q.816 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común,* más enmienda 1 (2001): *Perfil de servicios grupo de gestión de objetos*.

[5] Recomendación UIT-T Q.816 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común,* más enmienda 2 (2002): *Guía del usuario para la resolución de nombre local*.

[6] Recomendación UIT-T Q.816 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común* más corrigendum 1 (2001).

[7] Recomendación UIT-T Q.816.1 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común*.

[8] Recomendación UIT-T Q.822.1 (2001), *Servicio de gestión de la calidad de funcionamiento de la RGT basado en arquitectura de intermediario de petición de objeto común*.

[9] Recomendación UIT-T X.701 (1997), *Tecnología de la información – Interconexión de sistemas abiertos – Visión general de la gestión de sistemas*.

[10] Recomendación UIT-T X.720 (1992), *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Modelo de información de gestión*.

[11] Recomendación UIT-T X.721 (1992), *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Definición de la información de gestión*.

[12] Recomendación UIT-T X.722 (1992), *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Directrices para la definición de objetos gestionados*.

[13] Recomendación UIT-T X.723 (1993), *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Información de gestión genérica*.

[14] Recomendación UIT-T X.731 (1992), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de gestión de estados*.

[15] Recomendación UIT-T X.740 (1992), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de pista de auditoría de seguridad*.

[16] Recomendación UIT-T X.741 (1995), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Objetos y atributos para el control de acceso*.

[17] Recomendación UIT-T X.742 (1995), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de cómputo de utilización para contabilidad*.

[18] Recomendación UIT-T X.744 (1996), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de gestión de soporte lógico*.

[19] Recomendación UIT-T X.744 (1996), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de gestión de soporte lógico,* más corrigendum técnico 1 (1998).

[20] Recomendación UIT-T X.744 (1996), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de gestión de soporte lógico,* más corrigendum técnico 2 (2000).

[21] Recomendación UIT-T X.744 (1996), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de gestión de soporte lógico,* más corrigendum técnico 3 (2001).

[22] Recomendación UIT-T X.745 (1993), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de gestión de prueba*.

[23] Recomendación UIT-T X.746 (2000), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de planificación*.

[24] Recomendación UIT-T X.780 (2001), *Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común*.

[25] Recomendación UIT-T X.780 (2001), *Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común,* más corrigendum 1 (2001).

[26] Recomendación UIT-T X.780 (2001), *Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común,* más corrigendum 2 (2002).

[27] Recomendación UIT-T X.780 (2001), *Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común,* más enmienda 1 (2002)*, Objetos de sistema y guía del usuario para recuperación de atributos en bloque*.

[28] Recomendación UIT-T X.780.1 (2001), *Directrices de la RGT para la definición de interfaces de objetos gestionados mediante arquitectura de intermediario de petición de objeto común de la granularidad gruesa*.

[29] Recomendación UIT-T X.780.1 (2001), *Directrices de la RGT para la definición de interfaces de objetos gestionados mediante arquitectura de intermediario de petición de objeto común de la granularidad gruesa,* más enmienda 1 (2002)*, Fachadas de sistemas y orientaciones al usuario para la recuperación de atributos en bloque*.

[30] Recomendación UIT-T X.780.1 (2001), *Directrices de la RGT para la definición de interfaces de objetos gestionados mediante arquitectura de intermediario de petición de objeto común de la granularidad gruesa,* más corrigendum 1 (2002).

3 Definiciones

En esta Recomendación se utilizan los términos siguientes definidos en otras Recomendaciones.

3.1 En esta Recomendación se utilizan los términos siguientes definidos en la Rec. UIT-T M.3010 [1]:

a) Modelo de información de gestión.

b) Gestor.

3.2 En esta Recomendación se utiliza el término siguiente definido en la Rec. UIT-T X.701 [9]:

a) Clase de objeto gestionado.

3.3 En esta Recomendación se utilizan los términos siguientes definidos en la Rec. UIT-T X.744 [18]:

a) Copia de seguridad.

b) Entrega.

c) Ejecución.

d) Instalación.

e) Restablecimiento.

f) Reversión.

g) Utilización.

h) Validación.

3.4 En esta Recomendación se utilizan los términos siguientes definidos en la Rec. UIT-T X.780 [24]:

a) Jerarquía de herencia.

b) Árbol de denominación.

c) Objetos subordinados.

d) Objeto superior.

4 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

CORBA Arquitectura de intermediario de petición de objeto común (*common object request broker architecture*)

IDL Lenguaje de definición de interfaz (*interface definition language*)

MD Dispositivo de mediación (*mediation device*)

NE Elemento de red (*network element*)

OMG Grupo de gestión de objetos (*object management group*)

ORB Mediador de petición de objetos (*object request broker*)

OS Sistema de operaciones (*operations system*)

QA Adaptador Q (*Q adapter*)

RGT Red de gestión de las telecomunicaciones

UID Identificador universal (*universal identifier*)

5 Requisitos para la gestión de soporte lógico

La gestión de soporte lógico debe ser capaz de satisfacer los siguientes requisitos, a reserva de los posibles controles y condiciones impuestos:

a) ser capaz de pedir la entrega de soporte lógico a un sistema gestionado específico;

b) ser capaz de controlar la instalación de soporte lógico en un sistema gestionado, incluida la instalación de parches (por ejemplo, mejoras) y regresar a una versión anterior de soporte lógico;

c) ser capaz de iniciar la ejecución de un programa de soporte lógico;

d) ser capaz de indagar los atributos de todos los soportes lógicos mantenidos en un sistema gestionado;

e) ser capaz de crear y suprimir el soporte lógico mantenido en un sistema gestionado;

f) ser capaz de validar el soporte lógico mantenido en un sistema gestionado para comprobar su integridad;

g) ser capaz de restringir el uso de recursos de soporte lógico en un sistema gestionado para fines administrativos;

h) ser capaz de hacer una copia de seguridad de un elemento de soporte lógico y restablecer un elemento de soporte lógico copiado previamente.

El modelo no excluirá el uso de la gestión de registro, contabilidad, auditoría y licencia en cada estado de este proceso.

En todos los casos, el éxito o el fracaso de la operación tiene que ser informado al sistema gestor.

Existen relaciones entre objetos gestionados de soporte lógico que reflejan la manera en la cual dichos recursos utilizan otros recursos de soporte lógico. Esta relación se conoce como "utilización" y se emplea para indicar las versiones de soporte lógico que se han de utilizar en un momento dado. La gestión de esta relación queda en estudio.

6 Modelo para la función de gestión de soporte lógico

El objetivo principal de la función de gestión de soporte lógico es la gestión de piezas de soporte lógico. En este contexto, el soporte lógico incluye datos, información de control e instrucciones ejecutables. La visión de la gestión de soporte lógico se puede representar mediante la clase de objeto gestionado unidad de soporte lógico. Puesto que la gestión de soporte lógico que representa instrucciones ejecutables tiene características adicionales, se define una segunda clase de objeto gestionado, la unidad de soporte lógico ejecutable (que es una subclase de la unidad de soporte lógico).

Otro aspecto de la gestión de soporte lógico es la gestión de la entrega de soporte lógico al sistema gestionado. El soporte lógico no es gestionado necesariamente en las mismas unidades en las que se entrega. Por ejemplo, es posible que un cierto número de unidades diferentes de soporte lógico se entreguen juntas a un sistema gestionado (por ejemplo, en un disco compacto ROM). De manera similar, si una unidad de soporte lógico grande ha de ser entregada por una red de comunicaciones, quizás sea necesario dividirla en partes más pequeñas para la entrega. Otro caso es aquel en el que todo lo que se entrega son los cambios necesarios de un objeto de soporte lógico existente, por ejemplo, un parche. La gestión de la entrega se hace en términos del objeto gestionado distribución de soporte lógico.

6.1 Capacidades de gestión de soporte lógico

6.1.1 Creación

En un sistema gestionado se puede crear una unidad de soporte lógico utilizando la operación creación. Una unidad de soporte lógico puede ser creada en el estado creado o bien en el estado entregado. Una unidad de soporte lógico puede ser creada también como consecuencia de una operación entrega en un objeto gestionado distribución de soporte lógico o de alguna acción local. Como resultado de su creación, un objeto gestionado unidad de soporte lógico puede emitir una notificación de creación de objeto.

6.1.2 Supresión

Una unidad de soporte lógico puede ser suprimida de un sistema gestionado utilizando la operación supresión. Un efecto secundario de la operación supresión en una unidad de soporte lógico puede ser la eliminación de sus recursos subyacentes. Como resultado de su supresión, un objeto gestionado unidad de soporte lógico puede emitir una notificación de supresión de objeto.

6.1.3 Entrega

Se puede solicitar la entrega de un conjunto coordinado de unidades de soporte lógico. El resultado de la entrega indica el éxito o el fracaso. La entrega se realiza enviando la operación entrega al objeto gestionado distribución de soporte lógico. El resultado de una operación de entrega consiste en la entrega de una copia de los elementos de soporte lógico objetivo al sistema objetivo en el estado entregado. Un efecto secundario puede ser la creación de uno o más objetos asociados (por ejemplo, unidades de soporte lógico).

La agrupación en lotes de las unidades de soporte lógico y la elección del mecanismo de transferencia es un asunto local y está fuera del ámbito de la presente Recomendación. Por ejemplo, esta información puede ser preconfigurada o especificada en la operación de entrega junto con cualquier otra información asociada.

6.1.4 Ejecución de programa

Un sistema gestor puede solicitar la ejecución de soporte lógico ejecutable mediante la operación ejecución de programa. La presente Recomendación no especifica cómo puede ser gestionada seguidamente esa ejecución de soporte lógico ejecutable; simplemente proporciona un mecanismo de iniciación de dicha ejecución.

6.1.5 Obtención

Es posible obtener información sobre el soporte lógico (qué soporte lógico está presente, qué soporte lógico está disponible para su utilización, relaciones entre soportse lógicos, etc.) utilizando la operación obtención. Si la operación tiene éxito, el resultado de la misma contiene la información solicitada.

6.1.6 Instalación

La instalación adapta el soporte lógico para su utilización. Esto puede requerir, quizás, una cantidad considerable de procesamiento y de tiempo, e implicar la comprobación de que todas las partes de la versión de soporte lógico objetivo están presentes en el sistema gestionado (si han sido entregadas como parte de una actualización o ya están presentes) y el ensamblado de las mismas para que estén listas para su utilización. La instalación se realiza mediante la acción de instalación dirigida al objeto gestionado unidad de soporte lógico, del que es una característica opcional.

Un parche es una modificación de soporte lógico y puede ser representado por un objeto gestionado unidad de soporte lógico. Por consiguiente, un parche puede ser entregado, instalado, utilizado, copiado, etc., empleando la gestión de soporte lógico.

Un caso especial de instalación es el de un "parche" cuando lo que se entrega es el parche y la aplicación del parche es la instalación y produce una versión actualizada de soporte lógico listo para su utilización. La versión actualizada recibe un número de versión actualizada para identificación, entregado con el parche. El resultado de la instalación indica su éxito o fracaso.

Cuando la entrega de soporte lógico está completa, puede comenzar la instalación. La instalación quizás tenga que ser coordinada, pero la coordinación está fuera del alcance de la presente Recomendación. Ejemplos de coordinación son:

– la compleción y la entrega de varios elementos de soporte lógico antes de que otro elemento de soporte lógico pueda ser instalado;
– la comprobación de que el soporte lógico vigente está fuera de uso antes de habilitar uno nuevo;
– la comprobación de que versiones particulares de otro soporte lógico están (o no están) instaladas;
– la sincronización de la instalación con otros sistemas abiertos, permitiendo la instalación simultánea pero no acoplada en más de un sistema abierto, y proporcionando una manera de que las instalaciones en más de un sistema abierto se agrupen en una sola unidad de trabajo, etc.

6.1.7 Reversión

La operación reversión se utiliza para hacer que un objeto gestionado unidad de soporte lógico instalado vuelva a la situación de no instalado o para provocar la eliminación de uno o más parches aplicados. La operación reversión se realiza mediante una acción dirigida al objeto gestionado unidad de soporte lógico, del que es una característica opcional.

Puede ser necesario mantener información adicional sobre cómo proceder a la reversión del objeto gestionado unidad de soporte lógico, pero esto está fuera del alcance de la presente Recomendación.

6.1.8 Fijación del estado administrativo

Una vez que el soporte lógico ha sido adaptado para su utilización puede ponerse a disposición para su uso mediante la operación fijación del estado administrativo a desbloqueado, dirigida al objeto gestionado unidad de soporte lógico. Es posible retirar un soporte lógico especificado de un estado de disponibilidad fijando el estado administrativo del objeto gestionado unidad de soporte lógico a cierre o bloqueando.

6.1.9 Validación

La integridad de soporte lógico se puede comprobar utilizando la operación validación. También es posible validar qué soporte lógico entregado anteriormente permanece en condiciones de utilización. El resultado de la validación indica si el soporte lógico fue validado. La operación validación se realiza mediante una acción dirigida al objeto gestionado unidad de soporte lógico, del que es una característica opcional.

6.1.10 Notificaciones

Todas las operaciones aplicables a una unidad de soporte lógico pueden requerir que se envíen los resultados al sistema gestor junto con la confirmación de que la operación se ha completado. También puede ser necesario notificar a otro sistema gestor la compleción de estas operaciones cuando la petición de la operación no se hubiera originado en ese sistema gestor.

6.1.11 Copia de seguridad

La operación copia de seguridad puede ser utilizada por un sistema gestor para solicitar la realización de una copia de un objeto fijado como objetivo. Cuando se aplica la operación copia de seguridad a un objeto gestionado de soporte lógico, se hace una copia de los recursos subyacentes. No tiene efecto directo en los recursos de soporte lógico original.

6.1.12 Restablecimiento

La operación restablecimiento puede ser utilizada por un sistema gestor para solicitar que se efectúe el restablecimiento de un objeto objetivo copiado por seguridad previamente.

6.2 Relaciones entre objetos gestionados

La figura 1 contiene la jerarquía de herencia para los objetos gestionados gestión de soporte lógico y la figura 2 contiene la jerarquía de árbol de denominación de gestión de soporte lógico. Obsérvese que los objetos gestionados sistema pueden tener otras vinculaciones de nombre definidas que con un objeto gestionado elemento gestionado.

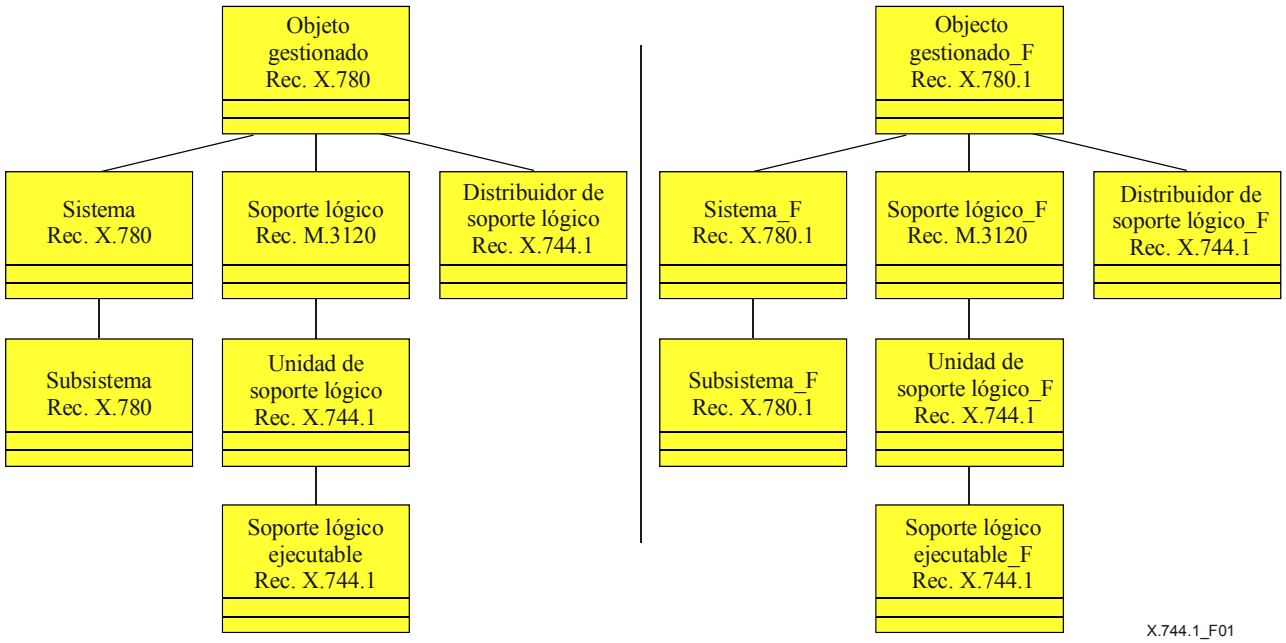

Figura 1/X.744.1 – Jerarquía de herencia de gestión de soporte lógico

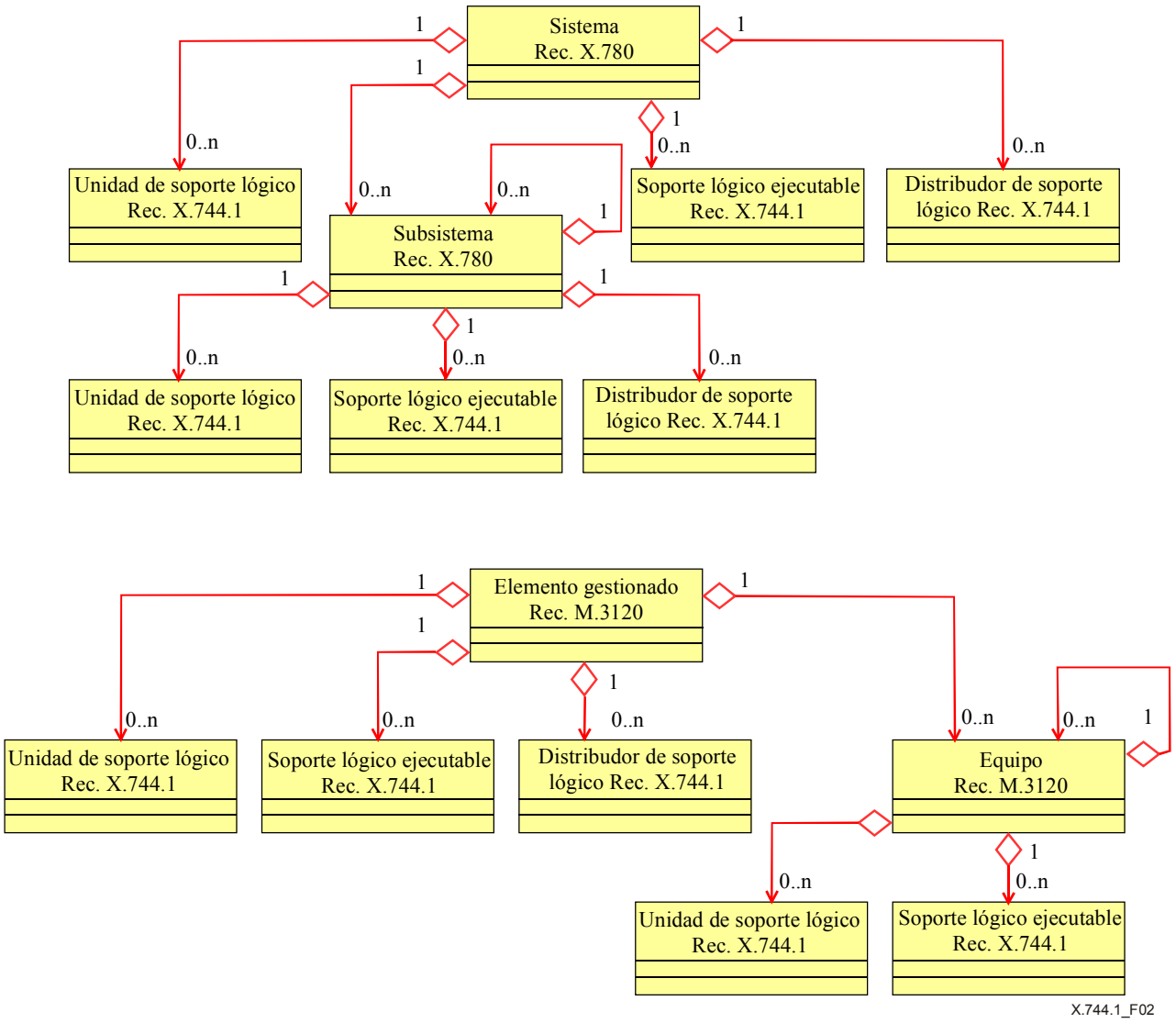

Figura 2/X.744.1 – Jerarquías de árbol de denominación de gestión de soporte lógico

6.3 Objeto gestionado unidad de soporte lógico

6.3.1 Creación de objetos gestionados unidad de soporte lógico

Cuando un objeto gestionado unidad de soporte lógico es creado por primera vez, representa el recurso que está preparado para aceptar la entrega de una pieza de soporte lógico.

Hay tres mecanismos u operaciones mediante las cuales puede ser creado un objeto gestionado unidad de soporte lógico:

1) Reproducción de soporte lógico en un sistema gestionado objetivo sin ninguna operación (por ejemplo, por medios locales).

2) Utilización de la operación creación para crear una unidad de soporte lógico en el estado creado o entregado.

3) Utilización de la operación entrega dirigida al objeto gestionado distribución de soporte lógico en el sistema fuente. Esto puede dar lugar a la creación de objetos gestionados unidad de soporte lógico en el destino objetivo en el estado entregado. La operación entrega aplicada a un objeto gestionado distribución de soporte lógico hace que el soporte lógico sea entregado a un destino objetivo. Cuando todo el soporte lógico requerido ha sido

entregado de satisfactoriamente, el objeto gestionado distribución de soporte lógico emite una notificación de resultado de entrega para señalar que la entrega se ha completado.

La operación creación origina la creación de un objeto gestionado unidad de soporte lógico en el destino objetivo. El éxito de la creación vendría indicado por la existencia de un objeto gestionado unidad de soporte lógico en el destino objetivo (se puede emitir también una notificación de creación de objeto).

Si el objeto gestionado unidad de soporte lógico fue creado previamente pero no entregado, puede emitir una notificación de cambio de valor de atributo cuando la entrega se ha completado, indicando que su estado interno ha cambiado de "creado" a "entregado". El objeto gestionado unidad de soporte lógico también puede ser creado en el estado entregado.

Una vez que el soporte lógico está en el estado entregado, la siguiente etapa es que el soporte lógico sea instalado. La instalación conlleva la adaptación de soporte lógico para su utilización. Por ejemplo, la instalación puede requerir la preparación y adaptación de la unidad de soporte lógico para una mejora, tomando quizás una copia de los recursos de soporte lógico vigente y aplicando los cambios a la copia. La instalación puede incluir el establecimiento de relaciones de configuración y dependencia entre la unidad de soporte lógico y otros objetos de soporte lógico.

Una vez instalado, el soporte lógico puede ser utilizado por otros objetos gestionados. Puede ser necesario ponerlo en disposición de uso fijando el estado administrativo a desbloqueado, si se halla en estado bloqueado. Esto se hace utilizando la operación fijación de estado administrativo en el atributo estado administrativo.

Se consigue que el soporte lógico no esté disponible para utilización bloqueándolo administrativamente, (es decir, fijando el estado administrativo a bloqueado o cierre). De esta manera se impide que algún proceso nuevo utilice el soporte lógico. Si el estado administrativo se fija a bloqueado, todos los procesos en curso en ese momento son abortados. Si el estado administrativo se fija a cierre, se deja que continúe la ejecución de los procesos en curso en ese momento pero no se permiten nuevos procesos. Cuando todos los procesos en curso que utilizan este soporte lógico han concluido su ejecución, el estado administrativo se fija automáticamente a bloqueado. Esto se hace utilizando la operación fijación de estado administrativo en el atributo estado administrativo.

La operación reversión se puede utilizar para invertir el efecto de una operación instalación anterior, o eliminar un parche aplicado.

La operación validación hace que se compruebe la integridad de soporte lógico, invocando quizás un algoritmo que genere una suma de control, verificando la existencia o no de virus, etc.

El respaldo de soporte lógico utilizando la operación copia de seguridad se considera equivalente a efectuar una copia de soporte lógico en curso para su utilización en caso de fallo de la versión existente. El soporte lógico copiado por seguridad puede ser restablecido utilizando la operación restablecimiento.

Obsérvese que puede ser necesario que el atributo facultativo parches aplicados tenga que ser mantenido internamente por el sistema gestionado cuando no está provisto en la clase objetos gestionados. Los parches aplicados deben ser mantenidos si se soportan los servicios instalación o reversión (incluso si el atributo parches aplicados no está en la clase objeto gestionado.

6.3.2 Estados del objeto gestionado unidad de soporte lógico

Un objeto gestionado unidad de soporte lógico puede estar en uno de los diversos estados del ciclo vital, dependiendo de la última operación que se realizó en él. Los estados posibles (y sus significados) son:

– Creado – La entrega del producto de soporte lógico no se ha completado, aunque algunos recursos arbitrarios del sistema gestionado han sido adjudicados a la unidad de soporte lógico.
– Entregado – La unidad de soporte lógico ha sido entregada satisfactoriamente al objeto gestionado.
– Instalado – El soporte lógico ha sido instalado satisfactoriamente en el sistema gestionado.

El cuadro 1 establece la correspondencia entre los estados de un objeto gestionado unidad de soporte lógico y los valores de estado y situación definidos en la Rec. UIT-T. X.731 [14].

Cuadro 1/X.744.1 – Estados internos de la unidad de soporte lógico

Estado de la unidad de soporte lógico	Valor {inicialización requerida} de situación de procedimiento	Valor {no instalado} de situación de disponibilidad	Estado operacional resultante
Creado	Presente	Presente	Inhabilitado
Entregado	Ausente	Presente	Inhabilitado
Instalado	Ausente	Ausente	Inhabilitado o habilitado

Para el objeto gestionado unidad de soporte lógico, el estado administrativo, el estado operativo, la situación de procedimiento y la situación de disponibilidad son atributos de estado obligatorios, mientras que el estado de utilización es un atributo opcional.

Los estados internos creado, entregado e instalado se excluyen mutuamente, es decir, una unidad de soporte lógico debe estar solamente en uno de esos estados en un momento dado. Sólo se permiten las transiciones de estados indicadas en el cuadro 2. El cuadro 2 sólo trata de las operaciones que pueden afectar a los estados. Se excluyen otras operaciones que no originan un cambio de estado.

Cuadro 2/X.744.1 – Matriz de transición de estados internos de la unidad de soporte lógico

Matriz de transición de estado		Estado resultante			
		(No existente)	Creado	Entregado	Instalado
Estado inicial	(No existente)	No es posible el cambio de estado	Crear o medios locales[a)]	Creación, medios locales o entrega a distribuidor de soporte lógico[a)]	No es posible el cambio de estado
	Creado	Supresión	–	Medios locales[a)]	No es posible el cambio de estado
	Entregado	Supresión	No es posible el cambio de estado	–	Instalación o medios locales[a)]
	Instalado	Supresión	No es posible el cambio de estado	Reversión[a)]	Instalar, reversión o cualquier operación que no origine un cambio de estado (es decir, respaldo, validación y restablecimiento)[a)]

[a)] Depende del comportamiento del objeto o de los medios locales.

Con independencia de los estados creado, entregado e instalado, existen los estados en validación y en fallo.

Los estados en validación y en fallo son estados secundarios que pueden estar presentes además de los estados internos. El estado en validación corresponde con un valor de situación de disponibilidad que incluye el valor {en prueba}. Un objeto gestionado unidad de soporte lógico puede pasar al estado en validación, o salir del mismo, independientemente de los valores de sus otros estados. Por ejemplo, una unidad de soporte lógico puede estar en el estado en validación mientras que está en fallo (véase el cuadro 3).

Cuadro 3/X.744.1 – Estado en validación de la unidad de soporte lógico

Estado de la unidad de soporte lógico	Valor {en prueba} de situación de disponibilidad	Estado operacional resultante
En validación	Presente	Inhabilitado o habilitado

Una unidad de soporte lógico que esté en el estado creado, entregado o instalado, puede también estar en el estado en fallo. El estado en fallo corresponde con un valor de situación de disponibilidad que incluye el valor {en fallo}. Las causas específicas para pasar al estado de fallo y salir del mismo, es un asunto local. Un objeto gestionado unidad de soporte lógico puede pasar al estado en fallo, o salir del mismo, independientemente de los valores de sus otros estados. Por ejemplo, el efecto secundario de una operación validación puede ser una transición al estado en fallo mientras se está en el estado en validación, o una unidad de soporte lógico puede existir en el estado en fallo como resultado de una operación reversión (véase el cuadro 4).

Cuadro 4/X.744.1 – Estado en fallo de la unidad de soporte lógico

Estado de la unidad de soporte lógico	Valor {en fallo} de situación de disponibilidad	Estado operacional resultante
En fallo	Presente	Inhabilitado

6.3.3 Operaciones asociadas con el objeto gestionado unidad de soporte lógico

Hay varias operaciones asociadas con un objeto gestionado unidad de soporte lógico, que se utilizan para hacer que cambie de estado.

Dichas operaciones son:

– Copia de seguridad – Hace que el soporte lógico sea copiado por seguridad en el destino objetivo.

– Creación – Genera un nuevo objeto gestionado unidad de soporte lógico en el sistema gestionado.

– Supresión – Suprime el objeto gestionado unidad de soporte lógico del sistema gestionado y puede tener además el efecto de supresión de recursos asociados.

– Instalación – Prepara el soporte lógico para su utilización.

– Restablecimiento – Restablece el soporte lógico copiado por seguridad a partir del destino objetivo.

– Reversión – Invierte la aplicación de un parche o una instalación.

– Validación – Comprueba la integridad de soporte lógico.

Las operaciones creación y supresión corresponden con las operaciones normalizadas creación y supresión definidas en la Rec. UIT-T X.780 [24], mientras que las demás operaciones corresponden con acciones. Además, los valores de atributo pueden ser indagados y modificados utilizando las operaciones obtención y fijación definidas en la Rec. UIT-T X.780.

6.3.3.1 Servicio copia de seguridad

El servicio copia de seguridad es utilizado por un sistema gestor para pedir que se efectúe una copia de seguridad de la información representada por el ejemplar de objeto objetivo (es decir, el objeto gestionado que representa el soporte lógico que está siendo copiado para seguridad).

Este servicio utiliza la operación copia de seguridad en la clase de objeto gestionado unidad de soporte lógico. El parámetro destino de copia de seguridad indicará el destino en el que la información debe copiarse. Son posibles destinos los siguientes:

– Un objeto gestionado local – En este caso, la operación copia de seguridad se efectuará internamente en el sistema gestionado. La información será copiada en el objeto gestionado suministrado.

– Un sistema distante – En este caso, la información copiada por seguridad será transferida fuera de línea al sistema distante utilizando algún medio local.

Cabe señalar las siguientes excepciones:

– **ApplicationError** de la Rec. UIT-T X.780 [24].

– **NOinformationBackupPackage** – Si el lote copia de seguridad de información no es soportado en este ejemplar.

– **BackupSoftwareProcessingFailure** – Parámetro destino de copia de seguridad no válido para la operación copia de seguridad.

– **ConcurrentOperationRequestFailure** – Esta clase de objeto gestionado unidad de soporte lógico ya tiene una petición pendiente de la operación copia de seguridad o restablecimiento (u otra). Los criterios para determinar si se soportan o no peticiones concurrentes para una clase particular de objeto gestionado unidad de soporte lógico son específicos del sistema.

Como una petición de copia de seguridad puede tomar bastante tiempo, la operación copia de seguridad retornará inmediatamente. Se emitirá una notificación de informe de copia de seguridad tras la compleción de la copia de seguridad del objeto objetivo.

6.3.3.2 Servicio instalación

El servicio instalación es utilizado por un sistema gestor para dar instrucciones a un sistema gestionado de que instale un ejemplar de objeto unidad de soporte lógico entregado o instalado. Si es aplicable, el servicio instalación actualizará el valor del atributo parches aplicados.

El atributo soporte lógico objetivo indicará la fuente de soporte lógico que se ha de instalar. La fuente debe ser única, desde la perspectiva de parches aplicados. La fuente puede ser una o más de las siguientes:

– Id de parche – Identificador específico del sistema.

– Puntero de parche – Clase de objeto gestionado (o subclase) unidad de soporte lógico.

La operación instalación devolverá el valor del atributo parches aplicados del objeto unidad de soporte lógico al cual está dirigido el servicio.

Cabe señalar las siguientes excepciones:

– **ApplicationError** de la Rec. UIT-T X.780 [24].

– **NOinstallPackage** – Si el lote instalación no es soportado en este ejemplar.

– **InstallSoftwareProcessingFailure** – Parámetro de información de instalación no válido.

– **OperationStateMismatch** – Si la unidad de soporte lógico no puede ser instalada debido a una condición de estado no válida. Sólo los objetos gestionados unidad de soporte lógico en el estado entregado o instalado pueden ser instalados. Además, el estado operacional debe

estar habilitado, el estado administrativo debe estar desbloqueado y el estado de utilización debe estar activo o en reposo.

– **ConcurrentOperationRequestFailure** – Esta clase de objeto gestionado unidad de soporte lógico ya tiene una petición pendiente de la operación instalación o reversión (u otra). Los criterios de si se soportan o no peticiones concurrentes para una clase particular de objeto gestionado unidad de soporte lógico son específicos del sistema.

6.3.3.3 Servicio restablecimiento

El servicio restablecimiento es utilizado por un sistema gestor para pedir la realización de un restablecimiento en la información representada por el ejemplar de objeto fijado como objetivo. Esto restablecerá una copia de seguridad previa.

Este servicio utiliza la operación restablecimiento en la clase de objeto gestionado unidad de soporte lógico. La operación restablecimiento puede ser realizada independientemente del estado de la clase de objeto gestionado unidad de soporte lógico. El parámetro fuente de restablecimiento indicará el destino a partir del cual se restablecerá la información. Son posibles destinos los siguientes:

– Un objeto gestionado local – Un objeto gestionado local de la misma clase de aquel al cual se aplica esta operación. En este caso, la operación restablecimiento se efectuará internamente en el sistema gestionado.

– Un sistema distante – En este caso, la información de restablecimiento será transferida fuera de línea desde el sistema distante utilizando algún medio local.

El comportamiento del ejemplar de unidad de soporte lógico o el medio local determinará el estado de la unidad de soporte lógico cuando haya sido restablecida.

Cabe señalar las siguientes excepciones:

– **ApplicationError** de la Rec. UIT-T X.780 [24].

– **NOinformationRestorePackage** – Si el lote copia de seguridad de la información no es soportado en este caso.

– **RestoreSoftwareProcessingFailure** – Parámetro fuente de recurso no válido de la operación restablecimiento.

– **ConcurrentOperationRequestFailure** – Esta clase de objeto gestionado unidad de soporte lógico tiene ya una petición pendiente de operación de copia de seguridad o de restablecimiento (u otra). Los criterios de si se soportan o no peticiones concurrentes para una clase particular de objeto gestionado unidad de soporte lógico son específicos del sistema.

Como una petición de restablecimiento puede tomar mucho tiempo, la operación restablecimiento retornará inmediatamente. Se emitirá una notificación informe de restablecimiento después de la compleción del restablecimiento del objeto objetivo.

6.3.3.4 Servicio reversión

El servicio reversión es utilizado por un sistema gestor (por ejemplo, un OS) para indicar a un sistema gestionado que vuelva a su estado original un parche o un conjunto de parches aplicados al soporte lógico representado por el objeto gestionado unidad de soporte lógico.

Este servicio utiliza la operación reversión en la clase de objeto gestionado unidad de soporte lógico. El parámetro información de reversión indicará el parche o los parches que han de ser desinstalados. Cada identificador de parche aplicado es una opción de un identificador específico del sistema o de un caso del objeto unidad de soporte lógico, dependiendo de los valores originalmente suministrados en anteriores operaciones de instalación en este ejemplar de objeto unidad de soporte lógico.

La operación reversión devolverá el valor del atributo parches aplicados del objeto unidad de soporte lógico al cual está dirigido el servicio.

Si el servicio reversión revierte satisfactoriamente todos los parches que hasta ese momento han sido instalados (es decir, el atributo parches aplicados está vacío), el estado interno de la unidad de soporte lógico cambiará de instalado a entregado.

Cabe señalar las siguientes excepciones:

– **ApplicationError** de la Rec. UIT-T X.780 [24].

– **NOrevertPackage** – Si el lote reversión no es soportado en este ejemplar.

– **OperationStateMismatch** – La unidad de soporte lógico está en un estado no válido para la operación reversión. Para ejecutar esta operación, la unidad de soporte lógico debe estar en el estado interno instalado, el estado administrativo debe estar desbloqueado, el estado operacional debe estar habilitado y el estado de utilización debe estar activo o en reposo.

– **RevertSoftwareProcessingFailure** – Parámetro de información de reversión no válido de la operación reversión.

– **ConcurrentOperationRequestFailure** – Esta clase de objeto gestionado unidad de soporte lógico tiene ya una petición pendiente de la operación instalación o reversión (u otra). Los criterios de si se soportan o no peticiones concurrentes para una clase particular de objeto gestionado unidad de soporte lógico son específicos del sistema.

6.3.3.5 Servicio validación

Este servicio utiliza la operación validación en la clase de objeto gestionado unidad de soporte lógico. El parámetro información de validación indicará el tipo de validación que se debe efectuar. Los posibles tipos son:

– Validación registrada – En este caso, la información de validación es proporcionada por otro objeto gestionado especificado.

– Validación por defecto – En este caso se utilizará la validación por defecto para este caso de objeto gestionado específico.

Cabe señalar las siguientes excepciones:

– **ApplicationError** de la Rec. UIT-T X.780 [24].

– **NOvalidationPackage** – Si el lote validación no es soportada en este ejemplar.

– **ValidateSoftwareProcessingFailure** – Uno o más argumentos no válidos de la operación validación.

– **OperationStateMismatch** – La unidad de soporte lógico está en un estado no válido para la operación validación. Para ejecutar la operación validación, la unidad de soporte lógico debe estar en el estado interno entregado o instalado.

– **ConcurrentOperationRequestFailure** – Esta clase de objeto gestionado unidad de soporte lógico tiene ya una petición pendiente de operación de validación (u otra). Los criterios para determinar si se soportan o no peticiones concurrentes para una clase particular de objeto gestionado unidad de soporte lógico son específicos del sistema.

Como una petición de validación puede tomar bastante tiempo, la operación validación retornará inmediatamente. Se emitirá una notificación de informe de validación después de la compleción de la validación del objeto objetivo.

6.3.4 Notificaciones sobre el objeto gestionado unidad de soporte lógico

El objeto gestionado unidad de soporte lógico tiene las siguientes notificaciones, incluidas las heredadas del objeto gestionado soporte lógico [2] (véase el cuadro 5):

Cuadro 5/X.744.1 – Notificaciones de unidad de soporte lógico

Notificación	Referencia	Lote condicional (si es condicional)
Cambio de valor de atributo	[24]	"itut_m3120::attributeValueChangeNotificationPackage"
Informe de copia de seguridad	6.3.4.1	"itut_x744d1::informationBackupPackage" and/or "itut_x744d1::informationAutoBackupPackage"
Creación de objeto	[24]	"itut_m3120::createDeleteNotificationsPackage"
Supresión de objeto	[24]	"itut_m3120::createDeleteNotificationsPackage"
Error de procesamiento	[24]	Mandatory
Informe de restablecimiento	6.3.4.2	"itut_x744d1::informationRestorePackage" and/or "itut_x744d1::informationAutoRestorePackage"
Cambio de estado	[24]	"itut_m3120::stateChangeNotificationPackage"
Informe de validación	6.3.4.3	"itut_x744d1::validationPackage"

Los cambios de los siguientes atributos (cuando se definen) originarán la emisión de notificaciones de cambio de valor de atributo (cuando se soportan):

– objetos afectados;
– situación de alarma;
– parches aplicados;
– estado de disponibilidad;
– lista de problemas vigentes;
– destino futuro de copia de seguridad automática;
– umbral de activación futura de copia de seguridad automática;
– restablecimiento automático permitido futuro;
– fuente de restablecimiento automático futuro;
– situación de procedimiento;
– etiqueta de usuario;
– versión.

Los cambios de los siguientes estados (cuando se definen) originarán la emisión de notificaciones de cambio de estado (cuando se soporten):

– estado operacional;
– estado administrativo;
– estado de utilización.

6.3.4.1 Notificación de informe de copia de seguridad

La notificación de informe de copia de seguridad se emite para informar una copia de seguridad de un objeto gestionado. La copia de seguridad puede haber sido iniciada automáticamente (de acuerdo con los criterios en los atributos umbral de activación futura de copia de seguridad automática y destino futuro de copia de seguridad automática), mediante petición de gestión (a través de la operación copia de seguridad) o iniciada por el sistema gestionado.

El destino de la copia de seguridad puede ser local (es decir, copia de seguridad a otro objeto unidad de soporte lógico dentro del sistema gestionado local) o fuera de línea a un sistema distante utilizando un protocolo de transferencia de ficheros particular (por ejemplo, FTAM). El resultado de la copia de seguridad será informado en esta notificación.

6.3.4.2 Notificación de informe de restablecimiento

La notificación de informe de restablecimiento se emite para informar un restablecimiento de un objeto gestionado a partir de una copia de seguridad previa. El restablecimiento puede haber sido iniciado automáticamente (de acuerdo con los atributos tipo de fuente de restablecimiento automático futuro y tipo de restablecimiento automático permitido futuro y los criterios específicos del sistema), mediante una petición de gestión (a través de la operación restablecimiento) o iniciado por el sistema de gestión.

La fuente de restablecimiento puede ser local (es decir, restablecimiento a partir de otro objeto unidad de soporte lógico dentro del sistema gestionado local) o fuera de línea a un sistema distante utilizando un protocolo de transferencia de ficheros particular (por ejemplo, FTAM). El resultado del restablecimiento será informado en esta notificación.

6.3.4.3 Notificación de informe de validación

La notificación de informe de validación se emite para informar los resultados de una operación de validación.

6.4 Objeto gestionado soporte lógico ejecutable

La clase de objeto gestionado soporte lógico ejecutable es una subclase de la clase de objeto gestionado unidad de soporte lógico (véase 6.3.1) con características adicionales, para describir su funcionalidad como ejecutable. El soporte lógico ejecutable representa el soporte lógico que puede ser ejecutado de alguna manera, por medios locales o a distancia, mediante la operación ejecución. Puede ser posible ejecutar el soporte lógico ejecutable con una instrucción de gestión, aunque también puede ser posible solamente ejecutar el soporte lógico mediante una acción local.

Aunque ello está fuera del alcance de la presente Recomendación, el modelo no impide que subclases de soporte ejecutable incluyan información, como la relativa a si el soporte lógico puede tener uno o varios usuarios, en qué condiciones el soporte lógico está activo u ocupado o si se permite un número máximo de usuarios.

6.4.1 Estados adicionales para el objeto gestionado soporte lógico ejecutable

El estado de utilización, definido en la Rec. UIT-T X.731 [14], indica si el soporte lógico ejecutable está siendo usado en ese momento. Para la clase de objeto gestionado soporte lógico ejecutable, el estado de utilización es un atributo de estado obligatorio, además de los ya obligatorios en el objeto gestionado unidad de soporte lógico (véase 6.3.2). En la clase de objeto gestionado soporte lógico ejecutable se permiten los valores de estado de utilización de reposo, activo y ocupado.

El valor de reposo del estado de utilización indica que no hay ejecuciones activas. El significado de los valores de activo y ocupado son específicos de la implementación y están fuera del alcance de la presente Recomendación. Por ejemplo, el valor de activo podrá significar que alguna ejecución está funcionando, mientras que el valor de ocupado significa que el programa ha alcanzado su capacidad máxima y cualquier otra petición de ejecución (utilizando el servicio ejecución de programa o por cualquier otro medio local) puede ser denegada o puesta en cola para ejecución ulterior. Las especializaciones pueden desear incluir un umbral del número de usuarios requeridos para alcanzar la capacidad máxima.

6.4.2 Operaciones adicionales para el objeto gestionado soporte lógico ejecutable

Las operaciones que siguen son operaciones en el objeto gestionado soporte lógico ejecutable, además de las definidas para el objeto gestionado unidad de soporte lógico (véase 6.3.3):

– Ejecución de programa – Esta operación hace que se ejecute el objeto gestionado de soporte lógico.

La operación ejecución de programa se utiliza para iniciar la ejecución del programa de soporte lógico representado por el soporte lógico ejecutable. El soporte lógico ejecutable debe estar instalado para que pueda ser ejecutado.

El comportamiento de soporte lógico ejecutable específico o el medio local determinará el estado de soporte lógico ejecutable cuando ha sido ejecutado.

6.4.2.1 Servicio ejecución de programa

El servicio ejecución de programa es utilizado por un sistema gestor para iniciar la ejecución de un programa representado por un objeto soporte lógico ejecutable.

El parámetro información adicional indicará los parámetros utilizados en la ejecución de soporte lógico.

Tras una iniciación satisfactoria, la respuesta de ejecución de programa contendrá:

– el Id de proceso;

– el propietario del proceso;

– el tiempo de arranque inicial;

– los parámetros de información adicional suministrados.

Cabe señalar las siguientes excepciones:

– **ApplicationError** de la Rec. UIT-T X.780 [24].

– **NOexecuteProgramPackage** – Si el lote ejecución de programa no es soportado en este ejemplar.

– **OperationStateMismatch** – El soporte lógico ejecutable está en un estado no válido para la operación ejecución de programa. Para efectuar la operación ejecución de programa, el soporte lógico ejecutable debe estar en el estado interno instalado, el estado administrativo debe estar desbloqueado, el estado operacional debe estar habilitado y el estado de utilización debe estar activo o en reposo.

– **ExecuteProgramSoftwareProcessingFailure** – Uno o más argumentos no válidos de la operación ejecución de programa.

6.5 Objeto gestionado distribución de soporte lógico

Un objeto gestionado distribución de soporte lógico es un objeto estático que representa el mecanismo o los mecanismos de entrega de un sistema gestionado. Se trata de un objeto gestionado que distribuye soporte lógico al sistema gestionado objetivo cuando recibe una operación entrega procedente del sistema gestor. Los parámetros de la operación entrega pueden ser utilizados para indicar el conjunto de soporte lógico que ha de entregarse, el destino objetivo para la entrega y el mecanismo de transferencia elegido. Aunque esta clase de objeto gestionado puede ser utilizada para iniciar la entrega mediante diversos mecanismos de transferencia, no modela ninguno de esos mecanismos de transferencia. Los modelos se dejan para las especializaciones.

Este objeto gestionado emite una notificación con los resultados de la distribución cuando se ha completado la misma.

6.5.1 Operaciones en el objeto gestionado distribución de soporte lógico

Las siguientes operaciones se efectúan en un objeto gestionado distribución de soporte lógico:

– Creación – Provoca la generación de un nuevo objeto gestionado distribución de soporte lógico. Se emitirá una notificación de creación de objeto.

– Entrega – Hace que el objeto gestionado distribución de soporte lógico efectúe la creación de soporte lógico especificado (por un método que está fuera del alcance de esta Recomendación) en el sistema gestionado objetivo y que cualesquiera recursos asociados se

asocien con los objetos gestionados unidad de soporte lógico que han de ser creados en el sistema gestionado objetivo como efecto secundario.

– Supresión – Hace que se suprima el objeto gestionado distribución de soporte lógico en el sistema gestionado. Se emitirá una notificación de supresión de objeto.

6.5.1.1 Servicio de entrega

El servicio de entrega es utilizado por un sistema gestor para pedir la distribución de soporte lógico o de un conjunto de soporte lógico. La información de la operación entrega identifica el soporte lógico que ha de ser distribuido. El resultado de una operación de entrega es que una copia de los elementos de soporte lógico objetivos es entregada al sistema objetivo en el estado interno entregado.

La agrupación en lotes de soporte lógico y la elección del mecanismo de transferencia es un asunto local y está fuera del alcance de la presente Recomendación. Por ejemplo, esta información puede estar preconfigurada o especificada en la operación entrega junto con cualquier otra información asociada.

El resultado de la compleción satisfactoria es que el soporte lógico que ha de ser distribuido es copiado al sistema objetivo, esto puede tener el efecto de la creación de objetos unidad de soporte lógico y/o soporte lógico ejecutable. Tras la compleción de la instrucción, se emite una notificación de resultado de entrega.

En la operación entrega se utilizan los siguientes parámetros:

– Id de entrega – Este parámetro opcional indica un identificador único para esta operación de entrega.

– Soporte lógico objetivo – Indica la fuente de soporte lógico que ha de ser entregado.

– Sistema objetivo – Indica el destino objetivo opcional para el soporte lógico que ha de ser entregado. Si no indica una destinación objetivo, el sistema utiliza medios locales para determinar el destino objetivo.

– Información de transferencia – Mecanismo de transferencia específico de la aplicación.

– Información adicional – Información adicional específica de la aplicación.

Cabe señalar las siguientes excepciones:

– **ApplicationError** de la Rec. UIT-T X.780 [24].

– **OperationStateMismatch** – El distribuidor de soporte lógico está en un estado no válido para la operación entrega. Para ejecutar la operación entrega, el distribuidor de soporte lógico debe estar en el estado administrativo desbloqueado y en el estado operacional habilitado.

– **DeliverSoftwareProcessingFailure** – Uno o más argumentos no válidos de la operación entrega.

6.5.2 Notificaciones sobre el objeto gestionado distribuidor de soporte lógico

El objeto gestionado distribuidor de soporte lógico tiene las siguientes notificaciones (véase el cuadro 6):

Cuadro 6/X.744.1 – Notificaciones del distribuidor de soporte lógico

Notificación	Referencia	Lote condicional (si hay condicional)
Resultado de entrega	6.5.2.1	Obligatorio
Creación de objeto	[24]	Obligatorio
Supresión de objeto	[24]	Obligatorio
Cambio de estado	[24]	Obligatorio

Los cambios en los siguientes estados harán que se emitan notificaciones de cambio de estado:

– Estado operacional.

– Estado administrativo.

6.5.2.1 Notificación de resultado de entrega

La notificación de resultado de entrega se emite desde el objeto gestionado cuando la operación de entrega es completada. Contiene los resultados finales de la operación y puede indicar la condición de éxito o de fracaso.

6.6 Utilización de identificadores universales (UID)

La presente Recomendación utiliza identificadores universales (UID, *universal identifiers*) definidos en la Rec. UIT-T X.780 [24]. Los identificadores universales tienen en cuenta las extensiones locales de los valores de constantes y de datos. Como ejemplos de dónde pueden ser utilizados cabe citar los argumentos definidos localmente para una operación ejecución de programa y para ampliar los tipos de ficheros enumerados.

Los identificadores universales utilizan los tipos de atributos tipo UID, tipo extensión de gestión y tipo de conjunto de información adicional. La sintaxis IDL para estos tipos (de la Rec. UIT-T X.780) es la siguiente:

```
struct UIDType {
  string moduleName; // módulo donde el valor es definido
  short value; // constante dentro del módulo
};

struct ManagementExtensionType {
  UIDType id; // identifica el tipo de info
  any info; // el tipo dependerá del id
};

typedef sequence <ManagementExtensionType> AdditionalInformationSetType;
```

Para hacer una extensión local de un identificador universal, los módulos de constantes de UID deben estar definidos en el nuevo IDL. Los módulos de constantes de UID contendrán una constante para cada constante soportada (con tipo de UID) o tipo de datos (con tipo de extensión de gestión y tipo de conjunto de información adicional). En 9.19 y 9.20 figuran ejemplos de módulos de constantes de UID utilizados en este modelo.

Para una descripción completa de los identificadores universales, véase la Rec. UIT-T X.780.

6.7 Relaciones

Se han identificado varias relaciones entre objetos gestionados de soporte lógico y también entre objetos gestionados de soporte lógico y otros objetos gestionados, que son las siguientes:

– Dependencia – Esta relación puede ser utilizada para modelar el hecho de que un objeto gestionado de soporte lógico depende de alguna manera de la presencia de otro objeto gestionado de soporte lógico. Esta relación se puede utilizar para modelar la aplicación de parches.

– Configuración – Esta relación puede ser utilizada para modelar el hecho de que un objeto gestionado unidad de soporte lógico puede afectar al comportamiento de otro objeto gestionado unidad de soporte lógico. Por ejemplo, un tipo de caracteres adicional podría ser modelado como una relación de configuración. Esta relación puede utilizarse también para modelar mejoras y la aplicación de parches.

– Utilización – Esta relación puede ser utilizada para mostrar qué otros objetos gestionados utilizan objetos gestionados de soporte lógico. Esos otros objetos gestionados son probablemente los que representan procesos en curso en el sistema gestionado.

El fallo de un objeto gestionado puede hacer que fallen también cualesquiera objetos gestionados que dependan de él. Sin embargo, no provocará el fallo de objetos gestionados con una relación de configuración (aunque su comportamiento puede cambiar). La detección de un soporte lógico averiado puede requerir la entrega e instalación de una nueva copia de soporte lógico o la entrega e instalación de una versión actualizada.

No obstante, la especificación de cualquiera de estas relaciones está fuera del alcance de la presente Recomendación, ya que las relaciones dependen de la aplicación.

7 Relación con otras funciones

Las siguientes funciones son proporcionadas por otras funciones de gestión de sistemas:

– calidad de funcionamiento de soporte lógico, tratada en la Rec. UIT-T Q.822.1 [8];

– rastreo de auditoría de soporte lógico, tratado por la función de rastreo de auditoría de la seguridad (véase la Rec. UIT-T X.740 [15]);

– soporte de la seguridad de soporte lógico, tratado por los objetos y atributos de control de acceso (véase la Rec. UIT-T X.741 [16]);

– contabilidad de la utilización de soporte lógico, tratada por la función de cómputo de la utilización para fines de contabilidad (véase la Rec. UIT-T X.742 [17]);

– prueba de soporte lógico (que incluye la instalación del entorno de prueba, ejecución de la prueba de soporte lógico, fijación de puntos de corte y suspensión y reanudación del entorno de prueba de soporte lógico), tratada por la función de gestión de pruebas (véase la Rec. UIT-T X.745 [22]);

– planificación de las funciones y operaciones de soporte lógico, tratada por la función de planificación (véase la Rec. UIT-T X.746 [23]).

8 Cumplimiento y conformidad

Esta cláusula define los criterios que deben ser satisfechos por otras normas que alegan cumplir la presente Recomendación y las funciones que deben ser ejecutadas por los sistemas que alegan conformidad con la presente Recomendación.

8.1 Conformidad de sistema

8.1.1 Puntos de conformidad

Esta cláusula describe los puntos de conformidad que deben ser soportados por los sistemas que alegan conformidad con estas especificaciones:

1) Una implementación que alega conformidad con estos requisitos debe:
   - Soportar:
     - el perfil de conformidad básico de la Rec. UIT-T Q.816 [3]. En este caso, cada ejemplar de objeto gestionado será un objeto CORBA ejemplificado; o
     - el perfil de conformidad básica de la Rec. UIT-T Q.816.1 [7]. En este caso, cada clase de objeto gestionado soportado tendrá un objeto CORBA de fachada ejemplificado (véanse las Recomendaciones UIT-T Q.816.1 y Q.780.1 [28]).
   - Soportar:
     - Los requisitos de la unidad de soporte lógico (sin el soporte de soporte lógico ejecutable o de distribuidor de soporte lógico).
     - Los requisitos de soporte lógico ejecutable (sin el soporte de unidad de soporte lógico o de distribuidor de soporte lógico).
     - Los requisitos del distribuidor de soporte lógico (sin el soporte de unidad de soporte lógico o de soporte lógico ejecutable). Esto puede ocurrir antes que el soporte lógico sea entregado.
     - Los requisitos de la unidad de soporte lógico y de soporte lógico ejecutable (sin el soporte de distribuidor de soporte lógico).
     - Los requisitos de unidad de soporte lógico y de distribuidor de soporte lógico (sin el soporte de soporte lógico ejecutable).
     - Los requisitos de soporte lógico ejecutable y del distribuidor de soporte lógico (sin el soporte de unidad de soporte lógico).
     - Los requisitos de la unidad de soporte lógico, de soporte lógico ejecutable y del distribuidor de soporte lógico.
   - Utilizar el IDL indicado en la cláusula 9.
2) Una implementación que alega conformidad con los requisitos de la unidad de soporte lógico debe:
   - Soportar el objeto gestionado unidad de soporte lógico especificado en 6.3.
   - Soportar la creación de al menos un objeto gestionado de la clase de objeto gestionado unidad de soporte lógico.
3) Una implementación que alega conformidad con los requisitos de soporte lógico ejecutable debe:
   - Soportar el objeto gestionado soporte lógico ejecutable especificado en 6.4.
   - Soportar la creación de al menos un objeto gestionado de la clase de objeto gestionado soporte lógico ejecutable.
4) Una implementación que alega conformidad con los requisitos del distribuidor de soporte lógico debe:
   - Soportar el objeto gestionado distribuidor de soporte lógico especificado en 6.5.
   - Soportar la creación de al menos un objeto gestionado de la clase de objeto gestionado distribuidor de soporte lógico.

8.2 Directrices de la declaración de conformidad

Los usuarios de este marco deben tener cuidado al redactar las declaraciones de conformidad. Como los módulos IDL se están utilizando como espacios de nombres, éstos pueden estar divididos a través de ficheros, como lo permiten las reglas IDL de OMG. De este modo, cuando un módulo es ampliado, su nombre no cambiará. En cambio, se añadirá simplemente un nuevo fichero IDL. Por consiguiente, la simple indicación del nombre de un módulo en una declaración de conformidad no bastará para identificar un conjunto de interfaces IDL. La declaración de conformidad debe identificar un documento y el año de publicación para asegurar que se identifica la versión correcta de IDL.

9 Listado IDL de la Rec. UIT-T X.744.1

```
#ifndef _itut_x744_1_idl_
#define _itut_x744_1_idl_

#include <itut_x780.idl>
#include <itut_x780_1.idl>
#include <itut_x780ct.idl>
#include <itut_m3120.idl>

#pragma prefix "itu.int"

/**
Este código IDL (que comienza con la línea "#ifndef … " hasta el fin de esta
sección) está destinado a ser almacenado en un fichero denominado
"itut_x744_1.idl" situado en el trayecto de búsqueda utilizado por el compilador
IDL en su sistema. Se debe utilizar un compilador que soporte la versión CORBA
especificada en la Rec. UIT-T Q.816.
*/

/**
Este módulo, itut_x744d1, contiene la definición de interfaces IDL para la Rec.
UIT-T X.744. Las definiciones IDL en este fichero son las interfaces de objetos.
*/

module itut_x744d1
{

/**
```

9.1 Importaciones

```
*/
     /**
     Tipos importados de la Rec. UIT-T X.780
     */

     typedef itut_x780::AdditionalInformationSetType AdditionalInformationSetType;
     typedef itut_x780::AdministrativeStateType AdministrativeStateType;
     typedef itut_x780::ApplicationErrorInfoType ApplicationErrorInfoType;
     typedef itut_x780::AvailabilityStatusSetType AvailabilityStatusSetType;
     typedef itut_x780::DeletePolicyType DeletePolicyType;
     typedef itut_x780::ExternalTimeType ExternalTimeType;
     typedef itut_x780::GeneralizedTimeType GeneralizedTimeType;
     typedef itut_x780::Istring Istring;
     typedef itut_x780::ManagementExtensionType ManagementExtensionType;
     typedef itut_x780::MONameType MONameType;
     typedef itut_x780::NameBindingType NameBindingType;
     typedef itut_x780::NullType NullType;
     typedef itut_x780::OperationalStateType OperationalStateType;
```

```
    typedef itut_x780::ProceduralStatusSetType ProceduralStatusSetType;
    typedef itut_x780::StringSetType StringSetType;
    typedef itut_x780::UIDType UIDType;
    typedef itut_x780::UsageStateType UsageStateType;

    /**
    Tipos importados de la Rec. UIT-T M.3120
    */

    typedef itut_m3120::AlarmStatusType AlarmStatusType;
    typedef itut_m3120::AlarmSeverityAssignmentProfileNameType
            AlarmSeverityAssignmentProfileNameType;
    typedef itut_m3120::ArcProbableCauseSetType ArcProbableCauseSetType;
    typedef itut_m3120::ArcIntervalProfileNameType ArcIntervalProfileNameType;
    typedef itut_m3120::ArcTimeType ArcTimeType;

/**
```

9.2 Declaraciones hacia adelante

```
*/
    /**
    Declaraciones hacia adelante de interfaces
    */

    interface ExecutableSoftware;
    interface ExecutableSoftware_F;
    interface ExecutableSoftwareFactory;
    interface SoftwareDistributor;
    interface SoftwareDistributor_F;
    interface SoftwareDistributorFactory;
    interface SoftwareUnit;
    interface SoftwareUnit_F;
    interface SoftwareUnitFactory;

    /**
    Declaraciones hacia adelante de valuetype
    */

    valuetype ExecutableSoftwareValueType;
    valuetype SoftwareDistributorValueType;
    valuetype SoftwareUnitValueType;

/**
```

9.3 Estructuras y typedefs

```
*/

    /**
    Patch ::= CHOICE {
      patchId   GraphicString, -- identificador específico de sistema --
      patchPointerObjectInstance } -- de la clase de objeto unidad de soporte
                                   -- lógico –
    AppliedPatches ::= SEQUENCE OF Patch
    */

    enum PatchChoice
    {
      patchIdChoice, // identificador específico de sistema
      patchPointerChoice // de la clase de objeto unidad de soporte lógico
                         // (o soporte lógico ejecutable)
    };
```

```
union PatchType switch (PatchChoice)
{
  case patchIdChoice:
     Istring patchId;
  case patchPointerChoice:
     MONameType patchPointer;
};

/**
Tipo de atributo Applied Patches
*/

typedef sequence <PatchType> AppliedPatchesSeqType;

/**
BackupDestination ::= CHOICE {
  localObject ObjectInstance,
  inLine NULL, -- en línea en la notificación en additionalInfo --
  offLine GraphicString -- sistema distante, por ejemplo, FTAM --}

La opción inLine no está soportada en la Rec. UIT-T X.744.1
*/

enum BackupDestinationChoice
{
  localObjectChoice,
  offLineChoice // sistema distante, por ejemplo, FTAM
};

union BackupDestinationType switch (BackupDestinationChoice)
{
  case localObjectChoice:
     MONameType localObject;
  case offLineChoice:
     Istring offLine;
};

/**
CheckSum ::= BIT STRING

Tipo de atributo Check Sum. El algoritmo para calcular la suma de control
se determina localmente
*/

typedef long CheckSumType;

/**
Tipo de atributo Date Of Creation
*/

typedef GeneralizedTimeType DateOfCreationType;

/**
Date ::= CHOICE {
  time   GeneralizedTime ,
  noSuchInformationNULL}
*/

enum DateChoice
{
  timeChoice,
  noSuchInformationChoice
};
```

```
union DateType switch (DateChoice)
{
  case timeChoice:
     GeneralizedTimeType time;
  case noSuchInformationChoice:
     NullType noInformation;
};

/**
Tipo de atributo File Location
*/

typedef sequence <Istring> FileLocationSetType;

/**
FileType ::= INTEGER{
  unstructuredText (0), -- FTAM-1
  unstructuredBinary (1), -- FTAM-3
  blockSpecial (2)}

El tipo de fichero ha sido convertido a UIDType, originalmente basado en el
módulo FileTypeConst. Esto permite que las aplicaciones añadan sus propios
tipos de ficheros

Tipo de atributo File Type
*/

typedef UIDType FileTypeType;

/**
Tipo de atributo Future Auto Backup Trigger Threshold

Obsérvese que el tipo flotante se utiliza para concordancia con los tipos
de umbral de la Rec. UIT-T Q.822.1, pero éste no es un verdadero
umbral Q.822.1
*/

typedef float FutureAutoBackupTriggerThresholdType;

/**
Tipo de atributo Future Auto Restore Allowed – TRUE significa que se
permite
*/

typedef boolean FutureAutoRestoreAllowedType;

/**
AutoRestoreSource ::= CHOICE {
  localObject ObjectInstance,
  remoteSystem GraphicString -- fuera de línea del sistema distante
}
*/

enum AutoRestoreSourceChoice
{
  autoRestoreSourceLocalObjectChoice,
  autoRestoreSourceRemoteSystemChoice // fuera de línea del sistema
                                      // distante
};
```

```
union AutoRestoreSourceType switch (AutoRestoreSourceChoice)
{
  case autoRestoreSourceLocalObjectChoice:
     MONameType localObject;
  case autoRestoreSourceRemoteSystemChoice:
     Istring offLine;
};

/**
Tipo de atributo Future Auto Restore Source
*/

typedef AutoRestoreSourceType FutureAutoRestoreSourceType;

/**
Tipo de atributo Identity Of Creator
*/

typedef Istring IdentityOfCreatorType;

/**
Tipo de atributo Identity Of Last Modifier
*/

typedef Istring IdentityOfLastModifierType;

/**
InformationSize ::=     CHOICE {
  numberOfBits[0] INTEGER,
  numberOfBytes    [1] INTEGER}
*/

enum InformationSizeChoice
{
  numberOfBitsChoice,
  numberOfBytesChoice
};

union InformationSizeType switch (InformationSizeChoice)
{
  case numberOfBitsChoice:
     long bits;
  case numberOfBytesChoice:
     long bytes;
};

/**
LastBackupDestination ::= CHOICE {
  notBackedUp NULL,
  localObject ObjectInstance,
  managingSystem AE-title,
  remoteSystem GraphicString}

La opción inLine no es soportada en la Rec. UIT-T X.744.1
*/

enum LastBackupDestinationChoice
{
  lastBackupDestinationLocalObjectChoice,
  lastBackupDestinationOffLineChoice,
  lastBackupDestinationNotBackedUpChoice
};
```

```
/**
Tipo de atributo Last Backup Destination
*/

union LastBackupDestinationType switch (LastBackupDestinationChoice)
{
  case lastBackupDestinationLocalObjectChoice:
     MONameType localObject;
  case lastBackupDestinationOffLineChoice:
     Istring offLine;
  case lastBackupDestinationNotBackedUpChoice:
     NullType noInformation;
};

/**
LastRestoreSource ::= CHOICE {
  notRestored NULL,
  localObject ObjectInstance,
  managingSystem AE-title,
  remoteSystem GraphicString}

La opción inLine no es soportada en la Rec. UIT-T X.744.1
*/

enum LastRestoreSourceChoice
{
  lastRestoreSourceLocalObjectChoice,
  lastRestoreSourceOffLineChoice,
  lastRestoreSourceNotRestoredChoice
};

/**
Tipo de atributo Last Restore Source
*/

union LastRestoreSourceType switch (LastRestoreSourceChoice)
{
  case lastRestoreSourceLocalObjectChoice:
     MONameType localObject;
  case lastRestoreSourceOffLineChoice:
     Istring offLine; // fuera de línea del sistema distante
  case lastRestoreSourceNotRestoredChoice:
     NullType noInformation;
};

/**
Tipo de atributo Note Field
*/

typedef Istring NoteFieldType;

/**
Tipo de atributo Date Delivered
*/

typedef DateType DateDeliveredType;

/**
Tipo de atributo Date Installed
*/

typedef DateType DateInstalledType;
```

```
/**
Tipo de atributo Date Of Last Modification
*/

typedef DateType DateOfLastModificationType;

/**
Tipo de atributo File Size
*/

typedef InformationSizeType FileSizeType;

/**
Tipo de atributo Future Auto Backup Destination
*/

typedef BackupDestinationType FutureAutoBackupDestinationType;

/**
Tipo de atributo Last Backup Time
*/

typedef DateType LastBackupTimeType;

/**
Tipo de atributo Last Restore Time
*/

typedef DateType LastRestoreTimeType;

/**
BackupResult ::= CHOICE {
  inLine [0]  CHOICE {
     successBIT STRING,
     fail-pduSizeLimitation  [3]  NULL,
     fail-securityLicensing  [4]  NULL,
     fail-unknown  [5]  NULL},
  local [1] SEQUENCE {
     destination ObjectInstance, -- en el sistema gestionado --
     éxito BOOLEAN -- TRUE para éxito --
     },
  offLine [2] SEQUENCE {
     destination GraphicString, -- el sistema distante
     result CHOICE {
      success  [6]  NULL,
      fail-securityLicensing [7]  NULL,
      fail-unknown  [8]  NULL}
}}

La opción inLine no es soportada en la Rec. UIT-T X.744.1
*/

enum BackupResultChoice
{
  backupResultFailureChoice, // ocurrió algún tipo de error mientras
                             // se copiaba el objeto para seguridad
  backupResultLocalChoice,
  backupResultOffLineChoice
};

union BackupResultType switch (BackupResultChoice)
{
  case backupResultFailureChoice:
     ApplicationErrorInfoType error; // de la Rec. UIT-T X.780
```

```
  case backupResultLocalChoice:
     MONameType localObject; // en el sistema gestionado
  case backupResultOffLineChoice:
     Istring offLine; // en el sistema gestionado
};

/**
DeliverId ::= CHOICE {
  globalValue OBJECT IDENTIFIER,
  localValue  INTEGER}
*/

typedef long DeliverIdType;

/**
DeliverIdTypeOpt es un tipo opcional. Si el discriminador es verdadero, el
valor está presente, en los demás casos el valor es NullType.
*/

union DeliverIdTypeOpt switch (boolean)
{
  case TRUE:
     DeliverIdType value;
  case FALSE:
     NullType noInformation;
};

/**
DeliverResult ::= INTEGER {
  pass (0),
  communicationError (1),
  equipmentError (2),
  qosError (3),
  accessDenied (4),
  notFound (5),
  insufficientSpace (6),
  alreadyDelivered (7),
  inProgress (8),
  unknown (9) }

El resultado de entrega ha sido convertido a UIDType, originalmente basado
en el módulo DeliverResultConst. Esto permite que las aplicaciones añadan
sus propios resultados de entrega.
*/

typedef UIDType DeliverResultType;

/**
Destination ::= CHOICE {
  single    AE-title,
  multiple  SET OF AE-title}
-- Obsérvese que la sintaxis de AE-title que se ha de utilizar es de la
-- enmienda 1 de la Rec. UIT-T X.227 | ISO CEI 8650-1 y no "ANY".
*/

typedef sequence <MONameType> MONameSetType;

enum DestinationChoice
{
  singleChoice,
  multipleChoice
};
```

```
union DestinationType switch (DestinationChoice)
{
  case singleChoice:
     MONameType single;
  case multipleChoice:
     MONameSetType multipleValues;
};

/**
ExecuteProgramReply ::= SEQUENCE {
  processId INTEGER,
  processOwner Identity,
  startTime GeneralizedTime,
  additionalInfo   SET OF ManagementExtension OPTIONAL }
*/

typedef Istring IdentityType;

struct ExecuteProgramReplyType
{
  long processId;
  IdentityType processOwner;
  GeneralizedTimeType startTime;
  AdditionalInformationSetType additionalInfo;
};

typedef AppliedPatchesSeqType InstallReplyType;

/**
¿Quién generó la petición?
*/

enum RequestType
{
  automaticRequest,
  managementRequest,    // es decir, método de ejecución
  managedSystemRequest
};

/**
¿Cuáles fueron los resultados de la operación restablecimiento?
*/

enum RestoreResultChoice
{
  restoreResultFailureChoice, // se produjo algún tipo de error al
                              // restablecer el objeto
  restoreResultLocalChoice,
  restoreResultOffLineChoice
};

union RestoreResultType switch (RestoreResultChoice)
{
  case restoreResultFailureChoice:
     ApplicationErrorInfoType error; // de la Rec. UIT-T X.780
  case restoreResultLocalChoice:
     MONameType localObject; // en el sistema gestionado
  case restoreResultOffLineChoice:
     Istring offLine; // en el sistema gestionado
};
```

```
/**
RestoreSource ::= CHOICE {
  localObject ObjectInstance,
  inLine BIT STRING,
  offLine GraphicString
     -- sistema distante por algún otro protocolo de transferencia,
     -- por ejemplo, FTAM --
}

La opción inline no es soportada en la Rec. UIT-T X.744.1
*/

enum RestoreSourceChoice
{
  restoreSourceLocalObjectChoice,
  restoreSourceOffLineChoice
     -- sistema distante por algún otro protocolo de transferencia,
     -- por ejemplo, FTAM --
};

union RestoreSourceType switch (RestoreSourceChoice)
{
  case restoreSourceLocalObjectChoice:
     MONameType localObject;
  case restoreSourceOffLineChoice:
     Istring offLine;
};

/**
RevertInfo ::= SEQUENCE OF CHOICE {
  patchId GraphicString, -- identificador específico de sistema --
  patchPointer ObjectInstance } -- clase de objeto soporte lógico
                                -- ejecutable -
*/

enum RevertChoice
{
  revertPatchIdChoice, // identificador específico de sistema
  revertPatchPointerChoice // clase de objeto (o subclase) de unidad
                           // de soporte lógico
};

union RevertType switch (RevertChoice)
{
  case revertPatchIdChoice:
     Istring patchId;
  case revertPatchPointerChoice:
     MONameType patchPointer;
};

typedef sequence <RevertType> RevertInfoSetType;

/**
RevertReply ::= SEQUENCE {
  revertedPatches  [0] AppliedPatches,
  additionalInfo   [1] SET OF ManagementExtension OPTIONAL }
*/

typedef AppliedPatchesSeqType RevertReplyType;

/**
DistributedSoftware ::= CHOICE {
  distibutedSoftwareId  GraphicString,
  distributedSoftwarePointer ObjectInstance }
*/
```

```
enum DistributedSoftwareChoice
{
  distributedSoftwareIdChoice,
  distributedSoftwarePointerChoice
};

union DistributedSoftwareType switch (DistributedSoftwareChoice)
{
  case distributedSoftwareIdChoice:
     Istring patchId; // identificador específico de sistema
  case distributedSoftwarePointerChoice:
     MONameType patchPointer; // clase de objeto (o subclase) de unidad
                              // de soporte lógico

};

typedef sequence <DistributedSoftwareType> TargetSoftwareSetType;

/**
Obsérvese que AdditionalInformationSetType es un conjunto
de ManagementExtensionType
*/

typedef ManagementExtensionType TransferInfoType;

/**
ValidateInfo ::= CHOICE {
  instanceDefaultValidationType  [0] NULL, -- asunto local --
  registeredValidationType   [1] OBJECT IDENTIFIER }
*/

enum ValidateInfoChoice
{
  registeredValidationTypeChoice,
  instanceDefaultValidationTypeChoice // asunto local
};

union ValidateInfoType switch (ValidateInfoChoice)
{
  case registeredValidationTypeChoice:
     MONameType instanceDefaultValidationType;
  case instanceDefaultValidationTypeChoice:
     NullType noInformation;
};

/**
ValidateReply ::= CHOICE {
  validationTerminated  [0] NULL,
  passValidation   [1] NULL,
  passValidationWithResult   [2] SET OF ManagementExtension,
  failValidation   [3] NULL,
  failValidationWithResult   [4] SET OF ManagementExtension }

La operación terminación de validación no es soportada en la
Rec. UIT-T X.744.1
*/

enum ValidateResultChoice
{
  passValidationWithResultChoice,
  failValidationWithResultChoice,
  passValidationChoice,
  failValidationChoice
};
```

```
union ValidateResultType switch (ValidateResultChoice)
{
  case passValidationWithResultChoice:
     AdditionalInformationSetType passValidationWithResult;
  case failValidationWithResultChoice:
     AdditionalInformationSetType failValidationWithResult;
  case failValidationChoice:
     ApplicationErrorInfoType error; // de la Rec. UIT-T X.780
  case passValidationChoice:
     NullType noInformation;
};

/**
BackupReply ::= SEQUENCE {
  reply  [0] CHOICE {
     success   NULL, -- para copia de seguridad local o fuera de línea
     inLine BIT STRING },
  additionalInfo   [1] SET OF ManagementExtension OPTIONAL }

En la Rec. UIT-T X.744.1 la operación copia de seguridad emite una
notificación de los resultados en vez de devolver los resultados
*/

/**
TerminateValidationInfo ::= ENUMERATED {
  cancel (0), -- descartar el resultado de la auditoría parcial --
  truncate (1) } -- informar el resultado de la auditoría parcialmente
                 -- completada –

TerminateValidationReply ::= CHOICE {
  noOutStandingValidation    [0] NULL,
  validationCancelled   [1] NULL,
  resultOfPartialValidation  [2] ValidateReply}

La operación terminación de validación no es soportada en
la Rec. UIT-T X.744.1
*/

/**
BackupDestinationTypeOpt es un tipo opcional. Si el discriminador es
verdadero, el valor está presente, en los demás casos el valor es NullType.
*/

union BackupDestinationTypeOpt switch (boolean)
{
  case TRUE:
     BackupDestinationType value;
  case FALSE:
     NullType noInformation;
};

/**
RestoreSourceTypeOpt es un tipo opcional. Si el discriminador es verdadero,
el valor está presente, en los demás casos el valor es NullType.
*/

union RestoreSourceTypeOpt switch (boolean)
{
  case TRUE:
     RestoreSourceType value;
  case FALSE:
     NullType noInformation;
};
```

```
/**
TargetSoftwareSetTypeOpt es un tipo opcional. Si el discriminador es
verdadero, el valor está presente, en los demás casos el valor es NullType.
*/

union TargetSoftwareSetTypeOpt switch (boolean)
{
  case TRUE:
     TargetSoftwareSetType value;
  case FALSE:
     NullType noInformation;
};

/**
RevertInfoSetTypeOpt es un tipo opcional. Si el discriminador es verdadero,
el valor está presente, en los demás casos el valor es NullType.
*/

union RevertInfoSetTypeOpt switch (boolean)
{
  case TRUE:
     RevertInfoSetType value;
  case FALSE:
     NullType noInformation;
};

/**
ValidateInfoTypeOpt es un tipo opcional. Si el discriminador es verdadero,
el valor está presente, en los demás casos el valor es NullType.
*/

union ValidateInfoTypeOpt switch (boolean)
{
  case TRUE:
     ValidateInfoType value;
  case FALSE:
     NullType noInformation;
};

struct ValidateSoftwareProcessingErrorType
{
  ValidateInfoTypeOpt validateInfo;
};

/**
DestinationTypeOpt es un tipo opcional. Si el discriminador es verdadero,
el valor está presente, en los demás casos el valor es NullType.
*/

union DestinationTypeOpt switch (boolean)
{
  case TRUE:
     DestinationType value;
  case FALSE:
     NullType noInformation;
};

/**
TransferInfoTypeOpt es un tipo opcional. Si el discriminador es verdadero,
el valor está presente, en los demás casos el valor es NullType.
*/
```

```
union TransferInfoTypeOpt switch (boolean)
{
  case TRUE:
     TransferInfoType value;
  case FALSE:
     NullType noInformation;
};

struct DeliverSoftwareProcessingErrorType
{
  DeliverIdTypeOpt deliverId;
  TargetSoftwareSetTypeOpt targetSoftware;
  DestinationTypeOpt targetSystem;
  TransferInfoTypeOpt transferInfo;
  AdditionalInformationSetType additionalInfo;
};

const string administrativeStatePackage =
  "itut_x744d1::administrativeStatePackage";
const string appliedPatchPackage = "itut_x744d1::appliedPatchPackage";
const string checkSumPackage = "itut_x744d1::checkSumPackage";
const string createDeleteNotificationsPackage =
  "itut_x744d1::createDeleteNotificationsPackage";
const string executeProgramPackage = "itut_x744d1::executeProgramPackage";
const string fileInformationPackage =
  "itut_x744d1::fileInformationPackage";
const string filePackage = "itut_x744d1::filePackage";
const string informationAutoBackupPackage =
  "itut_x744d1::informationAutoBackupPackage";
const string informationAutoRestorePackage =
  "itut_x744d1::informationAutoRestorePackage";
const string informationBackupPackage =
  "itut_x744d1::informationBackupPackage";
const string informationRestorePackage =
  "itut_x744d1::informationRestorePackage";
const string installPackage = "itut_x744d1::installPackage";
const string noteFieldPackage = "itut_x744d1::noteFieldPackage";
const string revertPackage = "itut_x744d1::revertPackage";
const string stateChangeNotificationPackage =
  "itut_x744d1::stateChangeNotificationPackage";
const string usageStatePackage = "itut_x744d1::usageStatePackage";
const string validationPackage = "itut_x744d1::validationPackage";

/**
```

9.4 Excepciones

```
*/

exception NOadministrativeStatePackage {};
exception NOappliedPatchPackage {};
exception NOcheckSumPackage {};
exception NOexecuteProgramPackage {};
exception NOfileInformationPackage {};
exception NOfilePackage {};
exception NOinformationAutoBackupPackage {};
exception NOinformationAutoRestorePackage {};
exception NOinformationBackupPackage {};
exception NOinformationRestorePackage {};
exception NOinstallPackage {};
exception NOnoteFieldPackage {};
exception NOrevertPackage {};
exception NOusageStatePackage {};
exception NOvalidationPackage {};
```

```
    exception BackupSoftwareProcessingFailure
    {
      BackupDestinationTypeOpt backupDestionation;
    };

    /**
    Utilizado por varios métodos para indicar que esta operación fracasó debido
    a otras operaciones pendientes o en ejecución
    */

    exception ConcurrentOperationRequestFailure {};

    exception RestoreSoftwareProcessingFailure
    {
      RestoreSourceTypeOpt attributes;
    };

    exception InstallSoftwareProcessingFailure
    {
      TargetSoftwareSetTypeOpt attributes;
    };

    exception RevertSoftwareProcessingFailure
    {
      RevertInfoSetTypeOpt attributes;
    };

    exception ValidateSoftwareProcessingFailure
    {
      ValidateSoftwareProcessingErrorType attributes;
    };

    exception ExecuteProgramSoftwareProcessingFailure
    {
      AdditionalInformationSetType additionalInfo;
    };

    exception DeliverSoftwareProcessingFailure
    {
      DeliverSoftwareProcessingErrorType attributes;
    };

    /**
    Utilizado por varios métodos para indicar que la operación no puede ser
    ejecutada debido a una condición de estado que no es válida para esta
    operación
    */

    exception OperationStateMismatch {};

/**
```

9.5 Unidad de soporte lógico

```
*/
    /**
    Este valuetype se utiliza para extraer todos los atributos
    */

    valuetype SoftwareUnitValueType : truncatable itut_m3120::SoftwareValueType
    {
      public AvailabilityStatusSetType availabilityStatus;
         // GET
```

```
public ProceduralStatusSetType proceduralStatus;
   // GET
public AppliedPatchesSeqType appliedPatches;
   // GET
   // appliedPatchPackage
public CheckSumType checkSum;
   // GET
   // checkSumPackage
public DateOfCreationType dateOfCreation;
   // GET
   // fileInformationPackage
public IdentityOfCreatorType identityOfCreator;
   // GET
   // fileInformationPackage
public DateOfLastModificationType dateOfLastModification;
   // GET
   // fileInformationPackage
public IdentityOfLastModifierType identityOfLastModifier;
   // GET
   // fileInformationPackage
public DateDeliveredType dateDelivered;
   // GET
   // fileInformationPackage
public DateInstalledType dateInstalled;
   // GET
   // fileInformationPackage
public FileLocationSetType fileLocation;
   // GET
   // filePackage
public FileSizeType fileSize;
   // GET
   // filePackage
public FileTypeType fileType;
   // GET
   // filePackage
public FutureAutoBackupTriggerThresholdType
          futureAutoBackupTriggerThreshold;
   // GET-REPLACE
   // informationAutoBackupPackage
public FutureAutoBackupDestinationType futureAutoBackupDestination;
   // GET-REPLACE
   // informationAutoBackupPackage
public FutureAutoRestoreSourceType futureAutoRestoreSource;
   // GET-REPLACE
   // informationAutoRestorePackage
public FutureAutoRestoreAllowedType futureAutoRestoreAllowed;
   // GET-REPLACE
   // informationAutoRestorePackage
public LastBackupTimeType lastBackupTime;
   // GET
   // informationBackupPackage
public LastBackupDestinationType lastBackupDestination;
   // GET
   // informationBackupPackage
public LastRestoreTimeType lastRestoreTime;
   // GET
   // informationRestorePackage
public LastRestoreSourceType lastRestoreSource;
   // GET
   // informationRestorePackage
public NoteFieldType noteField;
   // GET-REPLACE
   // noteFieldPackage
```

```
  public UsageStateType usageState;
     // GET
     // usageStatePackage

}; // valuetype SoftwareUnitValueType

/**
La clase de objeto unidad de soporte lógico es una clase de objetos
gestionados que proporciona información administrable asociada con soporte
lógico (esté en la forma de un fichero ejecutable, tal como un soporte
lógico de programa, o de un fichero no ejecutable, tal como una tabla de
correspondencia de datos o transconectada). El tipo de fichero, la
localización del fichero, y el tamaño del fichero están entre los atributos
identificados en esta clase de objeto. Cuando fileInformationPackage está
presente, el valor inicial obligatorio del atributo dateOfCreation es el
instante en que se crea el objeto gestionado. Si se han de soportar
operaciones de copia de seguridad, se han de soportar operaciones de
restablecimiento. Los lotes copia de seguridad de información y copia de
seguridad automática de información sólo existirán en los objetos
destinados unidad de soporte lógico que tengan también el lote
restablecimiento de información o el lote restablecimiento automático de
información. Los lotes restablecimiento de información y restablecimiento
automático de información sólo existirán en los objetos gestionados unidad
de soporte lógico que tengan también el lote copia de seguridad de
información o el lote copia de seguridad automática de información.
Cuando el lote de notificación de cambio de valores de atributos (heredado
de la superclase soporte lógico) está presente, se emitirá la notificación
attributeValueChange definida en la Rec. UIT-T X.780 cuando cambie el valor
de uno de los siguientes atributos:
- Objetos afectados
- Estado de alarma
- Parches aplicados
- Estado de disponibilidad
- Lista de problemas vigente
- Destino futuro de copia de seguridad automática
- Umbral futuro de activación de copia de seguridad automática
- Restablecimiento automático futuro permitido
- Fuente de restablecimiento automático futuro
- Estado de procedimiento
- Etiqueta de usuario
- Versión

Como cada uno de los atributos anteriores están en lotes condicionales,
el comportamiento para emitir la notificación attributeValueChange se
aplica solamente cuando los lotes condicionales correspondientes están
presentes en el objeto gestionado.
*/

/**
```

9.6 Interfaz SoftwareUnit

```
*/
     /**
     Objeto gestionado unidad de soporte lógico
     */

     interface SoftwareUnit : itut_m3120::Software
     {
       /**
       OperationalState y AdministrativeState del objeto gestionado soporte
       lógico no se requieren con el objeto gestionado unidad de soporte
       lógico. Esto significa que el atributo Packages para la unidad de
```

```
soporte lógico contendrá siempre la cadena
"itut_m3120::administrativeOperationalStatesPackage" string
*/

/**
La situación de disponibilidad se describe en la Rec. UIT-T X.731. El
comportamiento adicional figura en 6.3.2
*/

AvailabilityStatusSetType availabilityStatusGet ()
   raises (itut_x780::ApplicationError);

/**
La situación de procedimiento se describe en la Rec. UIT-T X.731. El
comportamiento adicional figura en 6.3.2
*/

ProceduralStatusSetType proceduralStatusGet ()
   raises (itut_x780::ApplicationError);

/**
Applied Patches identifica los parches que han sido aplicados y existen
aún en la unidad de soporte lógico que está representada por el objeto
unidad de soporte lógico. Los parches son actualizaciones de soporte
lógico. El valor de este atributo es de lectura solamente y es
actualizado automáticamente cuando se aplica un parche en el soporte
lógico. La sintaxis de este atributo es una secuencia de identificadores
de parches, donde un identificador de parche es una opción de un caso de
objeto (si el parche está representado por un objeto gestionado unidad
de soporte lógico) o una cadena gráfica (si el parche no está
representado por un objeto gestionado unidad de soporte lógico).

Obsérvese que el atributo opcional Applied Patches puede tener que ser
mantenido internamente cuando no está proporcionado en la clase de objeto
gestionado. Los parches aplicados deben ser mantenidos si se soportan los
servicios instalación o reversión (incluso si el atributo Applied Patches
no está en la clase de objetos gestionados).

PRESENT IF un ejemplar soporta parches de soporte lógico
*/

AppliedPatchesSeqType appliedPatchesGet ()
   raises (itut_x780::ApplicationError,
   NOappliedPatchPackage);

/**
Check Sum identifica la suma de control de la información de soporte
lógico representada por el objeto unidad de soporte lógico. PRESENT IF
un ejemplar soporta validación de suma de control
*/

CheckSumType checkSumGet ()
   raises (itut_x780::ApplicationError,
   NOcheckSumPackage);

/**
dateOfCreation indica el instante de creación del objeto gestionado. La
sintaxis de este atributo es del tipo GeneralizedTime. PRESENT IF un
ejemplar soporta información de fichero
*/

DateOfCreationType dateOfCreationGet ()
   raises (itut_x780::ApplicationError,
       NOfileInformationPackage);
```

```
/**
identityOfCreator identifica la entidad que crea el objeto gestionado.
Éste puede ser una cadena vacía si la identidad del creador no es
conocida. PRESENT IF un ejemplar soporta información de fichero
*/

IdentityOfCreatorType identityOfCreatorGet ()
   raises (itut_x780::ApplicationError,
       NOfileInformationPackage);

/**
dateOfLastModification identifica el instante de la última o más reciente
modificación (por ejemplo, parche, inversión, instalación, entrega) de
la información representada por el ejemplar objeto unidad de soporte
lógico. Los valores válidos para este atributo son GeneralizedTime o
NULL si la información no ha sido modificada. PRESENT IF un ejemplar
soporta información de fichero
*/

DateOfLastModificationType dateOfLastModificationGet ()
   raises (itut_x780::ApplicationError,
       NOfileInformationPackage);

/**
identityOfLastModifier identifica el último o más reciente modificador de
la información representada por el ejemplar objeto unidad de soporte
lógico. Puede ser una cadena vacía si no se conoce la identidad del
último modificador. PRESENT IF un ejemplar soporta información de fichero
*/

IdentityOfLastModifierType identityOfLastModifierGet ()
   raises (itut_x780::ApplicationError,
       NOfileInformationPackage);

/**
dateDelivered identifica el instante en que la información representada
por el ejemplar objeto unidad de soporte lógico fue entregado al sistema
gestionado. Los valores válidos para este atributo son GeneralizedTime o
NULL si la información no ha sido entregada. PRESENT IF un ejemplar
soporta información de fichero
*/

DateDeliveredType dateDeliveredGet ()
   raises (itut_x780::ApplicationError,
       NOfileInformationPackage);

/**
dateInstalled identifica el instante en que la información representada
por el ejemplar objeto unidad de soporte lógico fue instalado. Los
valores válidos para este atributo son GeneralizedTime o NULL si la
información no ha sido instalada. PRESENT IF un ejemplar soporta
información de fichero
*/

DateInstalledType dateInstalledGet ()
   raises (itut_x780::ApplicationError,
       NOfileInformationPackage);

/**
fileLocation especifica las direcciones completas (lógicas o físicas) del
objeto unidad de soporte lógico. El formato de la dirección depende de
la implementación, conforme con los convenios de direccionamiento de
ficheros del sistema gestionado en cuestión. Un conjunto vacío de este
```

```
atributo indica que la información a la cual se aplica el objeto
gestionado unidad de soporte lógico no ha sido aún instalada en el
sistema gestionado. PRESENT IF un ejemplar soporta representación de un
fichero
*/

FileLocationSetType fileLocationGet ()
   raises (itut_x780::ApplicationError,
       NOfilePackage);

/**
fileSize indica el tamaño del objeto gestionado unidad de soporte
lógico. PRESENT IF un ejemplar soporta representación de un fichero
*/

FileSizeType fileSizeGet ()
   raises (itut_x780::ApplicationError,
       NOfilePackage);

/**
fileType indica el tipo de la unidad de soporte lógico. Los posibles
tipos de unidades de soporte lógico son fichero binario no estructurado
(por ejemplo, fichero ejecutable), fichero de texto no estructurado
(por ejemplo, fichero no ejecutable), fichero especial de bloque, etc.
PRESENT IF un ejemplar soporta representación de un fichero
*/

FileTypeType fileTypeGet ()
   raises (itut_x780::ApplicationError,
       NOfilePackage);

/**
futureAutoBackupTriggerThreshold especifica el umbral que activará una
copia de seguridad automática para la información representada por el
ejemplar objeto. El umbral se define como el número de veces que la
información ha sido modificada. Cuando la información ha sido modificada
ese número de veces, se ejecutará una copia automática. El destino de la
copia de seguridad se especifica en el atributo
futureAutomacticBackupDestination. Estas copias de seguridad son
realizadas además de otra copia de seguridad periódica planificada. Al
completarse la copia de seguridad automática, se emitirá una
notificación backupReport del objeto. PRESENT IF un ejemplar soporta
copia de seguridad automática
*/

FutureAutoBackupTriggerThresholdType
       futureAutoBackupTriggerThresholdGet ()
   raises (itut_x780::ApplicationError,
       NOinformationAutoBackupPackage);

void futureAutoBackupTriggerThresholdSet
   (in FutureAutoBackupTriggerThresholdType
       futureAutoBackupTriggerThreshold)
   raises (itut_x780::ApplicationError,
       NOinformationAutoBackupPackage);

/**
futureAutoBackupDestination especifica el destino en el cual la
información representada por este ejemplar objeto será copiada para
seguridad. Los criterios de copia de seguridad se definen en el atributo
futureAutoBackupTriggerThreshold del objeto. El destino puede ser otro
ejemplar objeto de la misma clase de objeto si existe en el mismo sistema
gestionado local o en un sistema abierto distante (utilizando un
```

```
protocolo de transferencia de ficheros particular, por ejemplo FTAM).
PRESENT IF un ejemplar soporta copia de seguridad automática
*/

FutureAutoBackupDestinationType futureAutoBackupDestinationGet ()
   raises (itut_x780::ApplicationError,
       NOinformationAutoBackupPackage);

void futureAutoBackupDestinationSet
   (in FutureAutoBackupDestinationType futureAutoBackupDestination)
   raises (itut_x780::ApplicationError,
       NOinformationAutoBackupPackage);

/**
futureAutoRestoreSource especifica la fuente de la información que ha de
ser restablecida para la información representada por el ejemplar objeto
gestionado. La fuente es un objeto gestionado local o un sistema
distante. Los criterios de activación de un restablecimiento automático
de la información son específicos del sistema. PRESENT IF un ejemplar
soporta el restablecimiento automático
*/

FutureAutoRestoreSourceType futureAutoRestoreSourceGet ()
   raises (itut_x780::ApplicationError,
       NOinformationAutoRestorePackage);

void futureAutoRestoreSourceSet
   (in FutureAutoRestoreSourceType futureAutoRestoreSource)
   raises (itut_x780::ApplicationError,
       NOinformationAutoRestorePackage);

/**
futureAutoRestoreAllowed especifica si se permite el restablecimiento
automático de la información representada por este ejemplar objeto
gestionado. La sintaxis de este atributo es de tipo booleano con el valor
TRUE que significa permitido y FALSE que significa no permitido. Los
criterios que activan el restablecimiento de información automático son
específicos del sistema. PRESENT IF un ejemplar soporta restablecimiento
automático
*/

FutureAutoRestoreAllowedType futureAutoRestoreAllowedGet ()
   raises (itut_x780::ApplicationError,
       NOinformationAutoRestorePackage);

void futureAutoRestoreAllowedSet
   (in FutureAutoRestoreAllowedType futureAutoRestoreAllowed)
   raises (itut_x780::ApplicationError,
       NOinformationAutoRestorePackage);

/**
lastBackupTime identifica el instante de la última copia de seguridad de
la información representada por el ejemplar objeto gestionado. Los
valores válidos para este atributo son GeneralizedTime o NULL (si no se
ha ejecutado ninguna copia de seguridad de la información. PRESENT IF un
ejemplar soporta la operación copia de seguridad
*/

LastBackupTimeType lastBackupTimeGet ()
   raises (itut_x780::ApplicationError,
       NOinformationBackupPackage);
```

```
/**
lastBackupDestination identifica el destino, si existe, en el cual la
información representada por el objeto gestionado es copiada para
seguridad. PRESENT IF un ejemplar soporta la operación copia de seguridad
*/

LastBackupDestinationType lastBackupDestinationGet ()
   raises (itut_x780::ApplicationError,
       NOinformationBackupPackage);

/**
El servicio copia de seguridad es utilizado por un sistema gestor para
pedir que se efectúe una copia de seguridad de la información
representada por el ejemplar objeto objetivo (es decir, el objeto
gestionado que representa el soporte lógico que está siendo copiado para
seguridad. Tras la validación satisfactoria del argumento, la operación
copia de seguridad retornará inmediatamente. Se emitirá una notificación
de informe de copia de seguridad después de la compleción de la copia de
seguridad del objeto objetivo.

@param backupDestination  Indica el destino al cual se copiará la
                          información. Los posibles destinos son:
                          - Un objeto gestionado local – En este caso, la
                          operación copia de seguridad se ejecutará
                          internamente en el sistema gestionado. La
                          información será copiada para seguridad en el
                          ejemplar objeto gestionado suministrado.
                          - Opción fuera de línea – En este caso, la
                          información de copia de seguridad será
                          transferida fuera de línea al sistema distante
                          por algún medio local.
PRESENT IF un ejemplar soporta la operación copia de seguridad
*/

void backup
   (in BackupDestinationType backupDestionation)
   raises (itut_x780::ApplicationError,
       NOinformationBackupPackage,
       BackupSoftwareProcessingFailure,
       ConcurrentOperationRequestFailure);

/**
lastRestoreTime identifica el instante del último restablecimiento de la
información representada por el ejemplar objeto gestionado. Los valores
válidos para este atributo son GeneralizedTime o NULL (si no se ha
efectuado ningún restablecimiento de la información). PRESENT IF un
ejemplar soporta la operación restablecimiento o restablecimiento
automático
*/

LastRestoreTimeType lastRestoreTimeGet ()
   raises (itut_x780::ApplicationError,
       NOinformationRestorePackage,
       NOinformationAutoRestorePackage);

/**
lastRestoreSource identifica la fuente, si existe, a partir de la cual
se restablece la información representada por el objeto gestionado.
PRESENT IF un ejemplar soporta la operación restablecimiento o
restablecimiento automático
*/
```

```
LastRestoreSourceType lastRestoreSourceGet ()
   raises (itut_x780::ApplicationError,
       NOinformationRestorePackage,
       NOinformationAutoRestorePackage);

/**
El servicio restablecimiento es utilizado por un sistema gestor para
pedir que se efectúe el restablecimiento de la información representada
por el ejemplar objeto objetivo. Tras la validación satisfactoria del
argumento, la operación restablecimiento retornará inmediatamente. Se
emitirá una notificación de informe de restablecimiento tras la
compleción del restablecimiento del objeto objetivo.

@param restoreSource      Indica la fuente a la cual se restablecerá la
                          información. Las posibles fuentes son:
                          - Un objeto gestionado local de la misma clase
                          de aquel al que se aplica esta operación. En
                          este caso, la operación restablecimiento se
                          ejecutará internamente en el sistema
                          gestionado.
                          - Opción fuera de línea – En este caso,
                          la información de restablecimiento será
                          transferida fuera de línea desde el sistema
                          distante utilizando un protocolo de
                          transferencia de ficheros elegido localmente.
PRESENT IF un ejemplar soporta la operación restablecimiento
*/

void restore
   (in RestoreSourceType restoreSource)
   raises (itut_x780::ApplicationError,
       NOinformationRestorePackage,
       RestoreSoftwareProcessingFailure,
       ConcurrentOperationRequestFailure);

/**
El servicio de instalación es utilizado por un sistema gestor para
indicar a un sistema gestionado que instale un ejemplar objeto unidad de
soporte lógico entregado. Si es aplicable, el servicio de instalación
actualizará el valor del atributo Applied Patches.

@param targetSoftware     Indica la fuente de soporte lógico que ha de
                          ser instalado. La fuente debe ser única, desde
                          una perspectiva deApplied Patches. La fuente
                          puede ser una o más de las siguientes:
                          - Patch Id – Identificador específico del
                          sistema.
                          - Patch Pointer - Clase de objeto gestionado
                          unida desoporte lógico (o soporte lógico
                          ejecutable).
@return                   El servicio de instalación retornará
                          automáticamente el valor del atributo Applied
                          Patches del objeto unidad de soporte lógico al
                          cual está dirigido el servicio
PRESENT IF un ejemplar soporta la operación instalación
*/

InstallReplyType install
   (in TargetSoftwareSetType targetSoftware)
   raises (itut_x780::ApplicationError,
       NOinstallPackage,
       InstallSoftwareProcessingFailure,
       OperationStateMismatch,
       ConcurrentOperationRequestFailure);
```

```
/**
noteField contiene cualquier información o comentarios asociados con el
objeto gestionado que incluyen instrucciones específicas de instalación,
parámetros y valores de arranque, información necesaria para activar las
características del objeto gestionado, etc. PRESENT IF un ejemplar lo
soporta
*/

NoteFieldType noteFieldGet ()
   raises (itut_x780::ApplicationError,
       NOnoteFieldPackage);

void noteFieldSet
   (in NoteFieldType noteField)
   raises (itut_x780::ApplicationError,
       NOnoteFieldPackage);

/**
El servicio reversión es utilizado o un sistema gestor (por ejemplo, OS)
para indicar a un sistema gestionado que revierta un parche o conjunto de
parches aplicados de soporte lógico representado por el objeto
gestionado unidad de soporte lógico. Si el servicio reversión revierte
satisfactoriamente todos los parches que hasta ese momento habían sido
instalados (es decir, el atributo Applied Patches está vacío), el estado
interno de la unidad de soporte lógico cambiará de instalado a
entregado.

@param revertInfo          El parámetro Revert Information indicará uno o
                           más parches previamente aplicados que serán
                           desinstalados. Cada identificador de parche
                           aplicado es una opción de un identificador
                           específico del sistema o un ejemplar del
                           objeto unidad de soporte lógico, dependiendo
                           del valor originalmente suministrado en una
                           operación instalación previa en este ejemplar
                           objeto unidad de soporte lógico.
@return                    El servicio reversión devolverá automáticamente
                           el valor del atributo Applied Patches del
                           ejemplar objeto unidad de soporte lógico al
                           cual está dirigido el servicio.
PRESENT IF un ejemplar lo soporta
*/

RevertReplyType revert
   (in RevertInfoSetType revertInfo)
   raises (itut_x780::ApplicationError,
       NOrevertPackage,
       OperationStateMismatch,
       RevertSoftwareProcessingFailure,
       ConcurrentOperationRequestFailure);

/**
Usage State se describe en la Rec. UIT-T X.731. PRESENT IF un ejemplar lo
soporta
*/

UsageStateType usageStateGet ()
   raises (itut_x780::ApplicationError,
       NOusageStatePackage);
```

```
/**
El servicio de validación es utilizado por un sistema gestor para pedir
que se efectúe una validación de la información representada por el
ejemplar objeto unidad de soporte lógico.

@param validateInfo        Indica el tipo de validación que se debe
                           efectuar. Los tipos posibles son:
                           – Validación registrada – En este caso, la
                           información de validación es proporcionada por
                           otro objeto gestionado especificado.
                           - Validación por defecto – En este caso, se
                           utilizará la validación por defecto para este
                           objeto gestionado específico.
PRESENT IF un ejemplar lo soporta
*/

void validate
   (in ValidateInfoType validateInfo)
   raises (itut_x780::ApplicationError,
       NOvalidationPackage,
       OperationStateMismatch,
       ValidateSoftwareProcessingFailure,
       ConcurrentOperationRequestFailure);

/**
El parámetro Alarm Effect on Service de la Rec. UIT-T X.744 ha sido
incorporado en los parámetros processingErrorAlarm.

La notificación de alarma de error de procesamiento en la clase de
objeto gestionado soporte lógico es entonces obligatoria.
*/

MANDATORY_NOTIFICATION(
   itut_x780::Notifications, processingErrorAlarm)

/**
La notificación de informe de copia de seguridad se emite para informar
una copia de seguridad de la información representada por este objeto. Es
iniciada por la operación copia de seguridad. El destino de la copia de
seguridad puede ser local (es decir, copia de seguridad a otro objeto
unidad de soporte lógico dentro del sistema gestionado local) o fuera de
línea a un sistema distante utilizando un protocolo de transferencia de
ficheros particular (por ejemplo, FTAM). El resultado de la copia de
seguridad, es decir, éxito o fracaso, será informado en esta
notificación. PRESENT IF un ejemplar soporta la operación copia de
seguridad o copia de seguridad automática
*/

CONDITIONAL_NOTIFICATION(
   itut_x744d1::Notifications, backupReport, informationBackupPackage)

CONDITIONAL_NOTIFICATION(
   itut_x744d1::Notifications, backupReport,
   informationAutoBackupPackage)

/**
La notificación de informe de restablecimiento se emite para informar
el restablecimiento de un objeto gestionado a partir de una copia de
seguridad previa. El restablecimiento puede haber sido iniciado
automáticamente (de acuerdo con los atributos tipo de fuente de
restablecimiento automático futuro y tipo de restablecimiento automático
futuro permitido y los criterios específicos del sistema), mediante la
petición de gestión (a través de la operación restablecimiento) o
iniciado por el sistema gestionado. La fuente de restablecimiento puede
```

```
      ser local (es decir, restablecimiento a partir de otro objeto unidad de
      soporte lógico dentro del sistema gestionado local) o fuera de línea
      desde un sistema distante utilizando un protocolo particular de
      transferencia de ficheros (por ejemplo, FTAM). El resultado del
      restablecimiento será informado en esta notificación. PRESENT IF
      un ejemplar soporta la operación restablecimiento o restablecimiento
      automático
      */

      CONDITIONAL_NOTIFICATION(
         itut_x744d1::Notifications, restoreReport,
         informationRestorePackage)

      CONDITIONAL_NOTIFICATION(
         itut_x744d1::Notifications, restoreReport,
         informationAutoRestorePackage)

      /**
      La notificación de informe de validación se emite para informar la
      validación de un objeto gestionado. El resultado de la operación de
      validación será informado en esta notificación. PRESENT IF un ejemplar
      soporta la operación validación
      */

      CONDITIONAL_NOTIFICATION(
         itut_x744d1::Notifications, validateReport, validationPackage)

   }; // interface SoftwareUnit

/**
```

9.7 Interfaz SoftwareUnit_F

```
*/
   /**
   Objeto gestionado fachada de unidad de soporte lógico – véase la
   Rec. UIT-T X.780.1
   */

   interface SoftwareUnit_F : itut_m3120::Software_F
   {
     /**
     OperationalState y AdministrativeState del objeto gestionado soporte
     lógico se requieren ahora con el objeto gestionado unidad de soporte
     lógico. Esto significa que el atributo Packages para la unidad de
     soporte lógico contendrá siempre la cadena
     "itut_m3120::administrativeOperationalStatesPackage"
     */

     /**
     La situación de disponibilidad se describe en la Rec. UIT-T X.731. El
     comportamiento adicional figura en 6.3.2
     */

     AvailabilityStatusSetType availabilityStatusGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

     /**
     La situación de procedimiento se describe en la Rec. UIT-T X.731. El
     comportamiento adicional figura en 6.3.2
     */
```

```
ProceduralStatusSetType proceduralStatusGet
   (in MONameType name)
   raises (itut_x780::ApplicationError);

/**
Applied Patches identifica los parches que han sido aplicados y existen
aún en la unidad de soporte lógico que está representada por el ejemplar
objeto unidad de soporte lógico. Los parches son actualizaciones de
soporte lógico. El valor de este atributo es de lectura solamente y es
actualizado automáticamente cuando se efectúa un parche en el soporte
lógico. La sintaxis de este atributo es una secuencia de identificadores
de parches, donde un identificador de parche es una opción de un ejemplar
de objeto (si el parche está representado por un objeto gestionado unidad
de soporte lógico) o una cadena gráfica (si el parche no está
representado por un objeto gestionado unidad de soporte lógico).

Obsérvese que el atributo opcional Applied Patches puede tener que ser
mantenido internamente cuando no está proporcionado en la clase de objeto
gestionado. Los parches aplicados deben ser mantenidos si se soportan los
servicios instalación o reversión (incluso si el atributo Applied Patches
no está en la clase de objetos gestionados).

PRESENT IF un caso soporta la aplicación de parches de soporte lógico
*/

AppliedPatchesSeqType appliedPatchesGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOappliedPatchPackage);

/**
Check Sum identifica la suma de control de la información de soporte
lógico representada por el ejemplar objeto unidad de soporte lógico.
PRESENT IF un ejemplar soporta validación de suma de control
*/

CheckSumType checkSumGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
   NOcheckSumPackage);

/**
dateOfCreation indica el instante de creación del objeto gestionado. La
sintaxis de este atributo es del tipo GeneralizedTime. PRESENT IF un
ejemplar soporta información de ficheros
*/

DateOfCreationType dateOfCreationGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOfileInformationPackage);

/**
identityOfCreator identifica la entidad que crea el objeto gestionado.
Puede ser una cadena vacía si la identidad del creador no es conocida.
PRESENT IF un ejemplar soporta información de ficheros
*/

IdentityOfCreatorType identityOfCreatorGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOfileInformationPackage);
```

```
/**
dateOfLastModification identifica el instante de la última o más reciente
modificación (por ejemplo, parche, inversión, instalación, entrega) de la
información representada por el ejemplar objeto unidad de soporte
lógico. Los valores válidos para este atributo son GeneralizedTime o
NULL si la información no ha sido modificada. PRESENT IF un ejemplar
soporta información de ficheros
*/

DateOfLastModificationType dateOfLastModificationGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
      NOfileInformationPackage);

/**
identityOfLastModifier identifica el último o más reciente modificador de
la información representada por el ejemplar objeto unidad de soporte
lógico. Puede ser una cadena vacía si no se conoce la identidad del
último modificador. PRESENT IF un ejemplar soporta información de
ficheros
*/

IdentityOfLastModifierType identityOfLastModifierGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
      NOfileInformationPackage);

/**
dateDelivered identifica el instante en que la información
representada por el ejemplar objeto unidad de soporte lógico fue
entregada al sistema gestionado. Los valores válidos para este atributo
son GeneralizedTime o NULL si la información no ha sido entregada.
PRESENT IF un ejemplar soporta información de ficheros
*/

DateDeliveredType dateDeliveredGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
      NOfileInformationPackage);

/**
dateInstalled identifica el instante en que la información representada
por el ejemplar objeto unidad de soporte lógico fue instalada. Los
valores válidos para este atributo son GeneralizedTime o NULL si la
información no ha sido instalada. PRESENT IF un ejemplar soporta
información de ficheros
*/

DateInstalledType dateInstalledGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
      NOfileInformationPackage);

/**
fileLocation especifica las direcciones completas (lógicas o físicas) del
objeto unidad de soporte lógico. El formato de la dirección depende de
la implementación, conforme con los convenios de direccionamiento de
ficheros del sistema gestionado en cuestión. Un conjunto vacío de este
atributo indica que la información a la cual se aplica el objeto
gestionado unidad de soporte lógico no ha sido aún instalado en el
sistema gestionado. PRESENT IF un ejemplar soporta representación
de un fichero
*/
```

```
FileLocationSetType fileLocationGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOfilePackage);

/**
fileSize indica el tamaño del objeto gestionado unidad de soporte
lógico. PRESENT IF un ejemplar soporta representación de un fichero
*/

FileSizeType fileSizeGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOfilePackage);

/**
fileType indica el tipo de la unidad de soporte lógico. Los posibles
tipos de unidades de soporte lógico son fichero binario no estructurado
(por ejemplo, fichero ejecutable), fichero de texto no estructurado (por
ejemplo, fichero no ejecutable), fichero especial de bloques, etc.
PRESENT IF un ejemplar soporta representación de un fichero
*/

FileTypeType fileTypeGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOfilePackage);

/**
futureAutoBackupTriggerThreshold especifica el umbral que activará una
copia de seguridad automática para la información representada por el
ejemplar de objeto. El umbral se define como el número de veces que la
información ha sido modificada. Cuando la información ha sido modificada
ese número de veces, se ejecutará una copia automática. El destino de la
copia de seguridad se especifica en el atributo
futureAutomacticBackupDestination. Esta copias de seguridad son
realizadas además de otra copia de seguridad periódica planificada.
Una vez completada la copia de seguridad automática, se emitirá una
notificación backupReport del objeto. PRESENT IF un ejemplar soporta
copia de seguridad automática
*/

FutureAutoBackupTriggerThresholdType
       FutureAutoBackupTriggerThresholdGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOinformationAutoBackupPackage);

void futureAutoBackupTriggerThresholdSet
   (in MONameType name,
   in FutureAutoBackupTriggerThresholdType
       futureAutoBackupTriggerThreshold)
   raises (itut_x780::ApplicationError,
       NOinformationAutoBackupPackage);

/**
futureAutoBackupDestination especifica el destino en el cual la
información representada por este ejemplar de objeto será copiada para
seguridad. Los criterios de copia de seguridad se definen en el atributo
futureAutoBackupTriggerThreshold del ejemplar de objeto. El destino puede
ser otro ejemplar de objeto de la misma clase de objeto si existe en el
mismo sistema gestionado local o en un sistema abierto distante
(utilizando un protocolo de transferencia de ficheros particular, por
```

```
ejemplo FTAM). PRESENT IF un ejemplar soporta copia de seguridad
automática
*/

FutureAutoBackupDestinationType futureAutoBackupDestinationGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOinformationAutoBackupPackage);

void futureAutoBackupDestinationSet
   (in MONameType name,
   in FutureAutoBackupDestinationType futureAutoBackupDestination)
   raises (itut_x780::ApplicationError,
       NOinformationAutoBackupPackage);

/**
futureAutoRestoreSource especifica la fuente de la información que ha de
ser restablecida para la información representada por el ejemplar de
objeto gestionado. La fuente es un objeto gestionado local o un sistema
distante. Los criterios de activación de un restablecimiento automático
de la información son específicos del sistema. PRESENT IF un ejemplar
soporta el restablecimiento automático
*/

FutureAutoRestoreSourceType futureAutoRestoreSourceGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOinformationAutoRestorePackage);

void futureAutoRestoreSourceSet
   (in MONameType name,
   in FutureAutoRestoreSourceType futureAutoRestoreSource)
   raises (itut_x780::ApplicationError,
       NOinformationAutoRestorePackage);

/**
futureAutoRestoreAllowed especifica si se permite el restablecimiento
automático de la información representada por este ejemplar de objeto
gestionado. La sintaxis de este atributo es de tipo booleano con el
valor TRUE que significa permitido y FALSE que significa no permitido.
Los criterios que activan el restablecimiento de información automático
son específicos del sistema. PRESENT IF un ejemplar soporta
restablecimiento automático
*/

FutureAutoRestoreAllowedType futureAutoRestoreAllowedGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOinformationAutoRestorePackage);

void futureAutoRestoreAllowedSet
   (in MONameType name,
   in FutureAutoRestoreAllowedType futureAutoRestoreAllowed)
   raises (itut_x780::ApplicationError,
       NOinformationAutoRestorePackage);

/**
lastBackupTime identifica el instante de la última copia de seguridad de
la información representada por el ejemplar de objeto gestionado. Los
valores válidos para este atributo son GeneralizedTime o NULL (si no se
ha ejecutado ninguna copia de seguridad de la información). PRESENT IF un
ejemplar soporta la operación
*/
```

```
LastBackupTimeType lastBackupTimeGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOinformationBackupPackage);

/**
lastBackupDestination identifica el destino, si existe, en el cual la
información representada por el objeto gestionado es copiada para
seguridad. PRESENT IF un ejemplar soporta la operación copia de seguridad
*/

LastBackupDestinationType lastBackupDestinationGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOinformationBackupPackage);

/**
El servicio copia de seguridad es utilizado por un sistema gestor para
pedir que se efectúe una copia de seguridad de la información
representada por el ejemplar de objeto objetivo (es decir, el objeto
gestionado que representa el soporte lógico que está siendo copiado para
seguridad. Tras la validación satisfactoria del argumento, la operación
copia de seguridad retornará inmediatamente. Se emitirá una notificación
de informe de copia de seguridad después de la compleción de la copia de
seguridad del objeto objetivo.

@param name                Nombre del ejemplar de objeto gestionado
                           unidad de soporte lógico
@param backupDestination   Indica el destino al cual se copiará la
                           información. Los posibles destinos son:
                           - Un objeto gestionado local – En este caso,
                           la operación copia de seguridad se ejecutará
                           internamente en el sistema gestionado. La
                           información será copiada para seguridad en el
                           ejemplar de objeto gestionado suministrado.
                           - Opción fuera de línea – En este caso, la
                           información de copia de seguridad será
                           transferida fuera de línea al sistema distante
                           por algún medio local.
PRESENT IF un ejemplar soporta la operación copia de seguridad
*/

void backup
   (in MONameType name,
   in BackupDestinationType backupDestionation)
   raises (itut_x780::ApplicationError,
       NOinformationBackupPackage,
       BackupSoftwareProcessingFailure,
       ConcurrentOperationRequestFailure);

/**
lastRestoreTime identifica el instante del último restablecimiento de
la información representada por el ejemplar de objeto gestionado. Los
valores válidos para este atributo son GeneralizedTime o NULL (si no
se ha efectuado ningún restablecimiento de la información). PRESENT IF
un ejemplar soporta la operación establecimiento o restablecimiento
automático
*/

LastRestoreTimeType lastRestoreTimeGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOinformationRestorePackage,
       NOinformationAutoRestorePackage);
```

```
/**
lastRestoreSource identifica la fuente, si existe, a partir de la cual
la información representada por el objeto gestionado es restablecida.
PRESENT IF un ejemplar soporta la operación restablecimiento o
restablecimiento automático
*/

LastRestoreSourceType lastRestoreSourceGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOinformationRestorePackage,
       NOinformationAutoRestorePackage);

/**
El servicio restablecimiento es utilizado por un sistema gestor para
pedir que se restablezca la información representada por el ejemplar
objeto objetivo. Tras la validación satisfactoria del argumento, la
operación restablecimiento retornará inmediatamente. Se emitirá una
notificación informe de restablecimiento tras la completión del
restablecimiento del objeto objetivo.

@param name                Nombre del ejemplar de objeto gestionado unidad
                           de soporte lógico
@param restoreSource       Indica la fuente de la cual la información será
                           restablecida. Las posibles fuentes son:
                           - Un objeto gestionado local de la misma clase
                           que el objeto al que se aplica esta operación.
                           En este caso, la operación restablecimiento
                           será efectuada internamente en el sistema
                           gestionado.
                           - Opción fuera de línea – En este caso, la
                           información de restablecimiento será
                           transferida fuera de línea desde el sistema
                           distante utilizando un protocolo de
                           transferencia de ficheros elegido localmente.
PRESENT IF un ejemplar soporta la operación restablecimiento
*/

void restore
   (in MONameType name,
   in RestoreSourceType restoreSource)
   raises (itut_x780::ApplicationError,
       NOinformationRestorePackage,
       RestoreSoftwareProcessingFailure,
       ConcurrentOperationRequestFailure);

/**
El servicio de instalación es utilizado por un sistema gestor para
indicar a un sistema gestionado que instale un ejemplar de objeto unidad
de soporte lógico entregado. Si es aplicable, el servicio de instalación
actualizará el valor del atributo Applied Patches.

@param name                Nombre del ejemplar de objeto gestionado unidad
                           de soporte lógico
@param targetSoftware      Indica la fuente de soporte lógico que ha de
                           ser instalado. La fuente debe ser única, desde
                           una perspectiva de Applied Patches. La fuente
                           puede ser una o más de las siguientes:
                           - Patch Id – Identificador específico del
                           sistema.
                           - Patch Pointer – clase de objeto gestionado
                           unidad de soporte lógico (o soporte lógico
                           ejecutable)
```

```
@return                    El servicio de instalación retornará
                           automáticamente el valor del atributo Applied
                           Patches del ejemplar de objeto unidad de
                           soporte lógico al cual está dirigido el
                           servicio
PRESENT IF un ejemplar soporta la operación instalación
*/

InstallReplyType install
   (in MONameType name,
   in TargetSoftwareSetType targetSoftware)
   raises (itut_x780::ApplicationError,
       NOinstallPackage,
       InstallSoftwareProcessingFailure,
       OperationStateMismatch,
       ConcurrentOperationRequestFailure);

/**
noteField contiene cualquier información o comentarios asociados con el
objeto gestionado que incluyen instrucciones específicas de instalación,
parámetros y valores de arranque, información necesaria para activar las
características del objeto gestionado, etc. PRESENT IF un ejemplar lo
soporta
*/

NoteFieldType noteFieldGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOnoteFieldPackage);

void noteFieldSet
   (in MONameType name,
   in NoteFieldType noteField)
   raises (itut_x780::ApplicationError,
       NOnoteFieldPackage);

/**
El servicio reversión es utilizado o un sistema gestor (por ejemplo, OS)
para indicar a un sistema gestionado que revierta un parche o conjunto de
parches aplicados de soporte lógico representado por el objeto
gestionado unidad de soporte lógico. Si el servicio reversión revierte
satisfactoriamente todos los parches que hasta ese momento habían sido
instalados (es decir, el atributo Applied Patches está vacío), el estado
interno de la unidad de soporte lógico cambiará de instalado a
entregado.

@param name                Nombre del ejemplar de objeto gestionado unidad
                           de soporte lógico
@param revertInfo          El parámetro Revert Information indicará uno o
                           más parches previamente aplicados que serán
                           desinstalados. Cada identificador de parche
                           aplicado es una opción de un identificador
                           específico del sistema o un ejemplar de objeto
                           unidad de soporte lógico, dependiendo del
                           valor originalmente suministrado en una
                           operación instalación previa en este ejemplar
                           de objeto unidad de soporte lógico.
@return                    El servicio reversión devolverá automáticamente
                           el valor del atributo Applied Patches del
                           ejemplar de objeto unidad de soporte lógico al
                           cual está dirigido el servicio.
PRESENT IF un ejemplar lo soporta
*/
```

```
RevertReplyType revert
   (in MONameType name,
   in RevertInfoSetType revertInfo)
   raises (itut_x780::ApplicationError,
       NOrevertPackage,
       OperationStateMismatch,
       RevertSoftwareProcessingFailure,
       ConcurrentOperationRequestFailure);

/**
El estado de utilización se describe en la Rec. UIT-T X.731. PRESENT IF
un ejemplar lo soporta
*/

UsageStateType usageStateGet
   (in MONameType name)
   raises (itut_x780::ApplicationError,
       NOusageStatePackage);

/**
El servicio de validación es utilizado por un sistema gestor para pedir
que se efectúe una validación en la información representada por el
ejemplar de objeto unidad de soporte lógico.

@param name               Nombre del ejemplar de objeto gestionado unidad
                          de soporte lógico
@param validateInfo       Esto indica el tipo de validación que debe
                          efectuarse. Los tipos posibles son:
                          - Validación registrada – En este caso, la
                          información de validación es proporcionada
                          por otro objeto gestionado especificado.
                          - Validación por defecto – En este caso, se
                          utilizará la validación por defecto para este
                          ejemplar de objeto gestionado específico.
PRESENT IF un ejemplar lo soporta
*/

void validate
   (in MONameType name,
   in ValidateInfoType validateInfo)
   raises (itut_x780::ApplicationError,
       NOvalidationPackage,
       OperationStateMismatch,
       ValidateSoftwareProcessingFailure,
       ConcurrentOperationRequestFailure);

/**
El parámetro Alarm Effect on Service de la Rec. UIT-T X.744 ha sido
incorporado en los parámetros processingErrorAlarm.

La notificación de alarma de error de procesamiento en la clase de objeto
gestionado soporte lógico es ahora obligatoria.
*/

MANDATORY_NOTIFICATION(
   itut_x780::Notifications, processingErrorAlarm)

/**
La notificación de informe de copia de seguridad se emite para informar
una copia de seguridad de la información representada por este objeto. Es
iniciada por la operación copia de seguridad. El destino de la copia de
seguridad puede ser local (es decir, copia de seguridad a otro objeto
unidad de soporte lógico dentro del sistema gestionado local) o fuera de
línea a un sistema distante utilizando un protocolo de transferencia de
```

```
      ficheros particular (por ejemplo, FTAM). El resultado de la copia de
      seguridad, es decir, éxito o fracaso, será informado en esta
      notificación. PRESENT IF un ejemplar soporta la operación copia de
      seguridad o copia de seguridad automática
      */

      CONDITIONAL_NOTIFICATION(
         itut_x744d1::Notifications, backupReport, informationBackupPackage)

      CONDITIONAL_NOTIFICATION(
         itut_x744d1::Notifications, backupReport,
         informationAutoBackupPackage)

      /**
      La notificación de informe de restablecimiento se emite para informar
      el restablecimiento de un objeto gestionado a partir de una copia de
      seguridad previa. El restablecimiento puede haber sido iniciado
      automáticamente (de acuerdo con los atributos Future Auto Restore Source
      Type y Future Auto Restore Allowed Type y los criterios específicos del
      sistema), mediante la petición de gestión (a través de la operación
      restablecimiento) o iniciado por el sistema gestionado. La fuente de
      restablecimiento puede ser local (es decir, restablecimiento a partir
      de otro objeto unidad de soporte lógico dentro del sistema gestionado
      local) o fuera de línea desde un sistema distante utilizando un protocolo
      particular de transferencia de ficheros (por ejemplo, FTAM). El resultado
      del restablecimiento será informado en esta notificación. PRESENT IF
      un ejemplar soporta la operación restablecimiento o restablecimiento
      automático
      */

      CONDITIONAL_NOTIFICATION(
         itut_x744d1::Notifications, restoreReport,
         informationRestorePackage)

      CONDITIONAL_NOTIFICATION(
         itut_x744d1::Notifications, restoreReport,
         informationAutoRestorePackage)

      /**
      La notificación de informe de validación se emite para informar la
      validación de un objeto gestionado. El resultado de la operación de
      validación será informado en esta notificación. PRESENT IF un ejemplar
      soporta la operación validación
      */

      CONDITIONAL_NOTIFICATION(
         itut_x744d1::Notifications, validateReport, validationPackage)

   }; // interface SoftwareUnit_F

/**
```

9.8 Interfaz SoftwareUnitFactory

```
*/
   /**
   Creación y supresión para unidad de soporte lógico
   */

   interface SoftwareUnitFactory : itut_x780::ManagedObjectFactory
   {
     itut_x780::ManagedObject create
        (in NameBindingType nameBinding,
        in MONameType superior,
```

```
        in string reqID,   // autodenomnación si la cadena está vacía
        out MONameType name,
        in StringSetType packageNameList,
        in AdministrativeStateType adminstrativeState,
            // conditional
            // itut_m3120::administrativeOperationalStatePackage
            // GET-REPLACE
        in AlarmSeverityAssignmentProfileNameType profile,
            // conditional
            // itut_m3120::alarmSeverityAssignmentPointerPackage
            // GET-REPLACE
        in Istring userLabel,
            // conditional
            // itut_m3120::userLabelPackage
            // GET-REPLACE
        in Istring vendorName,
            // conditional
            // itut_m3120::vendorNamePackage
            // GET-REPLACE
        in Istring version,
            // conditional
            // itut_m3120::versionPackage
            // GET-REPLACE
        in ArcProbableCauseSetType arcProbableCauseList,
            // conditional
            // itut_m3120::arcPackage
            // GET-REPLACE, ADD-REMOVE
        in ArcIntervalProfileNameType arcIntervalProfilePointer,
            // conditional
            // itut_m3120::arcPackage
            // GET-REPLACE
        in ArcTimeType arcManagementRequestedInterval,
            // conditional
            // itut_m3120::arcPackage
            // GET-REPLACE
        in AvailabilityStatusSetType availabilityStatus,
            // GET
            // SET-BY-CREATE
        in FutureAutoBackupTriggerThresholdType
                      futureAutoBackupTriggerThreshold,
            // GET-REPLACE
            // informationAutoBackupPackage
        in FutureAutoBackupDestinationType futureAutoBackupDestination,
            // GET-REPLACE
            // informationAutoBackupPackage
        in FutureAutoRestoreSourceType futureAutoRestoreSource,
            // GET-REPLACE
            // informationAutoRestorePackage
        in FutureAutoRestoreAllowedType futureAutoRestoreAllowed,
            // GET-REPLACE
            // informationAutoRestorePackage
        in NoteFieldType noteField
            // GET-REPLACE
            // noteFieldPackage
        )
        raises (itut_x780::ApplicationError,
            itut_x780::CreateError);

    }; // interface SoftwareUnitFactory

/**
```

9.9 Soporte lógico ejecutable

```
*/
     /**
     Este valuetype se utiliza para extraer todos los atributos
     */

     valuetype ExecutableSoftwareValueType : truncatable SoftwareUnitValueType
     {
     }; // valuetype ExecutableSoftwareValueType

     /**
       La clase de objeto executableSoftware es una clase de objeto gestionado
     que proporciona información administrable asociada con un programa
     ejecutable en el sistema gestionado. El programa ejecutable real (que puede
     consistir en segmentos de código con o sin segmentos de datos, etc.) puede
     estar en un formato dependiente de máquina, no normalizado, que
     generalmente no es legible por el sistema gestor y el resto del mundo
     exterior. Se puede utilizar una operación denominada executeProgram
     (condicionalmente) para ejecutar el programa representado por el programa
     executableSoftware. El atributo usageState se utiliza para indicar si hay
     ejecuciones activas del programa.
     */

/**
```

9.10 Interfaz ExecutableSoftware

```
*/
     /**
     Objeto gestionado soporte lógico ejecutable
     */

     interface ExecutableSoftware : SoftwareUnit
     {
       /**
       UsageState del objeto gestionado unidad de soporte lógico se requiere
       con el objeto gestionado soporte lógico ejecutable. Esto significa que el
       atributo Packages para el soporte lógico ejecutable contendrá siempre la
       cadena "itut_x744d1::usageStatePackage"
       */

       /**
       El servicio ejecución de programa es utilizado por un sistema gestor para
       iniciar la ejecución de un programa representado por un objeto soporte
       lógico ejecutable. Para efectuar la operación ejecución de programa, el
       soporte lógico ejecutable debe estar en el estado interno instalado, el
       estado administrativo debe estar desbloqueado, el estado operacional debe
       estar habilitado y el estado de utilización debe estar activo o en
       reposo.

       @param additionalInfo      Esto define los parámetros utilizados en la
                                  ejecución de soporte lógico.
       @return                    Una petición satisfactoria será confirmada con
                                  información que incluye el ID de proceso,
                                  propietario del proceso, tiempo de comienzo
                                  inicial y los parámetros Additional Information
                                  suministrados.
       PRESENT IF un ejemplar lo soporta
       */
```

```
      ExecuteProgramReplyType executeProgram
         (in AdditionalInformationSetType additionalInfo)
         raises (itut_x780::ApplicationError,
             NOexecuteProgramPackage,
             OperationStateMismatch,
             ExecuteProgramSoftwareProcessingFailure);

    }; // interface ExecutableSoftware

/**
```

9.11 Interfaz ExecutableSoftware_F

```
*/
    /**
    Objeto gestionado fachada de unidad ejecutable – véase la
    Rec. UIT-T X.780.1
    */

    interface ExecutableSoftware_F : SoftwareUnit_F
    {
      /**
      UsageState del objeto gestionado unidad de soporte lógico se requiere
      con el objeto gestionado soporte lógico ejecutable. Esto significa que el
      atributo Packages para el soporte lógico ejecutable contendrá siempre
      la cadena "itut_x744d1::usageStatePackage"
      */

      /**
      El servicio ejecución de programa es utilizado por un sistema gestor para
      iniciar la ejecución de un programa representado por un objeto soporte
      lógico ejecutable. Para efectuar la operación ejecución de programa, el
      soporte lógico ejecutable debe estar en el estado interno instalado, el
      estado administrativo debe estar desbloqueado, el estado operacional debe
      estar habilitado y el estado de utilización debe estar activo o en
      reposo.

      @param name               Nombre del ejemplar de objeto gestionado
                                soporte lógico ejecutable
      @param additionalInfo     Define los parámetros utilizados en la
                                ejecución de soporte lógico.
      @return                   Una petición satisfactoria será confirmada con
                                información que incluye el ID de proceso,
                                propietario de proceso, tiempo de comienzo
                                inicial y los parámetros Additional Information
                                suministrados.
      PRESENT IF un ejemplar lo soporta
      */

      ExecuteProgramReplyType executeProgram
         (in MONameType name,
         in AdditionalInformationSetType additionalInfo)
         raises (itut_x780::ApplicationError,
             NOexecuteProgramPackage,
             OperationStateMismatch,
             ExecuteProgramSoftwareProcessingFailure);

    }; // interface ExecutableSoftware_F

/**
```

9.12 Interfaz ExecutableSoftwareFactory

```
*/
     /**
     Creación y supresión para la unidad ejecutable
     */

     interface ExecutableSoftwareFactory : itut_x780::ManagedObjectFactory
     {
       itut_x780::ManagedObject create
          (in NameBindingType nameBinding,
          in MONameType superior,
          in string reqID,   // autodenominación si la cadena está vacía
          out MONameType name,
          in StringSetType packageNameList,
          in AdministrativeStateType adminstrativeState,
              // conditional
              // itut_m3120::administrativeOperationalStatePackage
              // GET-REPLACE
          in AlarmSeverityAssignmentProfileNameType profile,
              // conditional
              // itut_m3120::alarmSeverityAssignmentPointerPackage
              // GET-REPLACE
          in Istring userLabel,
              // conditional
              // itut_m3120::userLabelPackage
              // GET-REPLACE
          in Istring vendorName,
              // conditional
              // itut_m3120::vendorNamePackage
              // GET-REPLACE
          in Istring version,
              // conditional
              // itut_m3120::versionPackage
              // GET-REPLACE
          in ArcProbableCauseSetType arcProbableCauseList,
              // conditional
              // itut_m3120::arcPackage
              // GET-REPLACE, ADD-REMOVE
          in ArcIntervalProfileNameType arcIntervalProfilePointer,
              // conditional
              // itut_m3120::arcPackage
              // GET-REPLACE
          in ArcTimeType arcManagementRequestedInterval,
              // conditional
              // itut_m3120::arcPackage
              // GET-REPLACE
          in AvailabilityStatusSetType availabilityStatus,
              // GET
              // SET-BY-CREATE
          in FutureAutoBackupTriggerThresholdType
              futureAutoBackupTriggerThreshold,
              // GET-REPLACE
              // informationAutoBackupPackage
          in FutureAutoBackupDestinationType futureAutoBackupDestination,
              // GET-REPLACE
              // informationAutoBackupPackage
          in FutureAutoRestoreSourceType futureAutoRestoreSource,
              // GET-REPLACE
              // informationAutoRestorePackage
          in FutureAutoRestoreAllowedType futureAutoRestoreAllowed,
              // GET-REPLACE
              // informationAutoRestorePackage
```

```
        in NoteFieldType noteField
            // GET-REPLACE
            // noteFieldPackage
        )
        raises (itut_x780::ApplicationError,
            itut_x780::CreateError);

    }; // interface ExecutableSoftwareFactory

/**
```

9.13 Distribuidor de soporte lógico

```
*/
    /**
    Este valuetype se usa para extraer todos los atributos
    */

    valuetype SoftwareDistributorValueType :
        truncatable itut_x780::ManagedObjectValueType
    {
      public AdministrativeStateType administrativeState;
        // GET-REPLACE
      public OperationalStateType operationalState;
        // GET

    }; // valuetype SoftwareDistributorValueType

    /**
    Un objeto gestionado distribuidor de soporte lógico es un objeto
    gestionado que distribuye soporte lógico al sistema gestionado objetivo
    cuando recibe una operación de entrega desde el sistema gestor. Este
    objeto gestionado notifica el resultado de la distribución cuando ésta se
    ha completado. Se emitirá stateChangeNotification definida en la
    Rec. UIT-T X.780 si cambia el valor del estado administrativo o del
    estado operacional.
    */

/**
```

9.14 Interfaz SoftwareDistributor

```
*/
    /**
    Objeto gestionado distribuidor de soporte lógico
    */

    interface SoftwareDistributor : itut_x780::ManagedObject
    {
      /**
      El estado administrativo se describe en la Rec. UIT-T X.731
      */

      AdministrativeStateType administrativeStateGet ()
        raises (itut_x780::ApplicationError);

      void administrativeStateSet
        (in AdministrativeStateType administrativeState)
        raises (itut_x780::ApplicationError);
```

```
/**
El estado operacional se describe en la Rec. UIT-T X.731
*/

OperationalStateType operationalStateGet ()
   raises (itut_x780::ApplicationError);

/**
El servicio de entrega es utilizado por un sistema gestor para pedir la
distribución de soporte lógico o de un conjunto de soporte lógico. La
información de la operación de entrega identifica el soporte lógico que
ha de ser distribuido. El resultado de una operación de entrega es que se
entrega una copia de los elementos de soporte lógico objetivo al sistema
objetivo en el estado interno entregado.

La agrupación de lotes de soporte lógico y la elección del mecanismo de
transferencia es un asunto local y está fuera del ámbito de la presente
Recomendación. Por ejemplo, esta información puede estar preconfigurada o
especificada en la operación entrega junto con cualquier otra información
asociada.

El resultado de la compleción satisfactoria es que el soporte lógico que
ha de ser distribuido es copiado al sistema objetivo; esto puede tener
el efecto de crear los objetos unidad de soporte lógico y/o soporte
lógico ejecutable. Tras la compleción de la instrucción, se emite una
notificación de resultados de entrega.

@param deliverId          Este parámetro opcional indica un identificador
                          único para esta operación de entrega.
@param targetSoftware     Indica la fuente de soporte lógico que ha de
                          ser entregado.
@param targetSystem       Indica el destino objetivo opcional de soporte
                          lógico que ha de ser entregado. Si no indica un
                          destino objetivo, el sistema utiliza medios
                          locales para determinar el destino objetivo.
@param transferInfo       Mecanismo de transferencia específico de la
                          aplicación
@param additionalInfo     Información adicional específica de la
                          aplicación.
*/

void deliver
   (in DeliverIdTypeOpt deliverId,
   in TargetSoftwareSetType targetSoftware,
   in DestinationType targetSystem,
   in TransferInfoType transferInfo,
   in AdditionalInformationSetType additionalInfo)
   raises (itut_x780::ApplicationError,
       OperationStateMismatch,
       DeliverSoftwareProcessingFailure);

/**
Se emite la notificación de resultado de entrega del objeto gestionado
cuando se completa la operación de entrega. Contiene los resultados
finales de la operación y puede indicar una condición de éxito o de
fracaso.
*/

MANDATORY_NOTIFICATION(
   itut_x744d1::Notifications, deliverResultNotification)
MANDATORY_NOTIFICATION(
   itut_x780::Notifications, objectCreation)
MANDATORY_NOTIFICATION(
   itut_x780::Notifications, objectDeletion)
```

```
      MANDATORY_NOTIFICATION(
         itut_x780::Notifications, stateChange)

   }; // interface SoftwareDistributor

/**
```

9.15 Interfaz SoftwareDistributor_F

```
*/
   /**
   Objeto gestionado fachada de distribuidor de soporte lógico – Véase la
   Rec. UIT-T X.780.1
   */

   interface SoftwareDistributor_F : itut_x780::ManagedObject_F
   {
     /**
     El estado administrativo se describe en la Rec. UIT-T X.731
     */

     AdministrativeStateType administrativeStateGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

     void administrativeStateSet
        (in MONameType name,
        in AdministrativeStateType administrativeState)
        raises (itut_x780::ApplicationError);

     /**
     El estado operacional se describe en la Rec. UIT-T X.731
     */

     OperationalStateType operationalStateGet
        (in MONameType name)
        raises (itut_x780::ApplicationError);

     /**
     El servicio de entrega es utilizado por un sistema gestor para pedir la
     distribución de soporte lógico o de un conjunto de soporte lógico. La
     información de la operación de entrega identifica el soporte lógico que
     ha de ser distribuido. El resultado de una operación de entrega es que se
     entrega una copia de los elementos de soporte lógico objetivo al sistema
     objetivo en el estado interno entregado.

     La agrupación de lotes de soporte lógico y la elección del mecanismo de
     transferencia es un asunto local y está fuera del ámbito de la presente
     Recomendación. Por ejemplo, esta información puede estar preconfigurada o
     especificada en la operación entrega junto con cualquier otra información
     asociada.

     El resultado de la compleción satisfactoria es que el soporte lógico que
     ha de ser distribuido es copiado al sistema objetivo; esto puede tener el
     efecto de crear los objetos unidad de soporte lógico y/o soporte lógico
     ejecutable. Tras la compleción de la instrucción, se emite una
     notificación de resultados de entrega.

     @param name              Nombre del caso de objeto gestionado
                              distribuidor de soporte lógico
     @param deliverId         Este parámetro opcional indica un identificador
                              único para esta operación de entrega.
```

```
      @param targetSoftware      Indica la fuente de soporte lógico que ha de
                                 ser entregado.
      @param targetSystem        Indica el destino objetivo opcional de soporte
                                 lógico que ha de ser entregado. Si no indica un
                                 destino objetivo, el sistema utiliza medios
                                 locales para determinar el destino objetivo.
      @param transferInfo        Mecanismo de transferencia específico de la
                                 aplicación.
      @param additionalInfo      Información adicional específica de la
                                 aplicación.
      */

      void deliver
         (in MONameType name,
         in DeliverIdTypeOpt deliverId,
         in TargetSoftwareSetType targetSoftware,
         in DestinationType targetSystem,
         in TransferInfoType transferInfo,
         in AdditionalInformationSetType additionalInfo)
         raises (itut_x780::ApplicationError,
             OperationStateMismatch,
             DeliverSoftwareProcessingFailure);

      /**
      Se emite la notificación de resultado de entrega del objeto gestionado
      cuando se completa la operación de entrega. Contiene los resultados
      finales de la operación y puede indicar una condición de éxito o de
      fracaso.
      */

      MANDATORY_NOTIFICATION(
         itut_x744d1::Notifications, deliverResultNotification)
      MANDATORY_NOTIFICATION(
         itut_x780::Notifications, objectCreation)
      MANDATORY_NOTIFICATION(
         itut_x780::Notifications, objectDeletion)
      MANDATORY_NOTIFICATION(
         itut_x780::Notifications, stateChange)

   }; // interface SoftwareDistributor_F

/**
```

9.16 Interfaz SoftwareDistributorFactory

```
*/
   /**
   Creación y supresión para distribuidor de soporte lógico
   */

   interface SoftwareDistributorFactory : itut_x780::ManagedObjectFactory
   {
      itut_x780::ManagedObject create
         (in NameBindingType nameBinding,
         in MONameType superior,
         in string reqID,   // auto naming if empty string
         out MONameType name,
         in StringSetType packageNameList,
         in AdministrativeStateType administrativeState
              // GET-REPLACE
         )
```

```
        raises (itut_x780::ApplicationError,
             itut_x780::CreateError);

    }; // interface SoftwareDistributorFactory

/**
```

9.17 Notificaciones

```
*/

    Notificaciones de interfaces
    {
      /**
      La notificación de informe de copia de seguridad se emite para informar
      una copia de seguridad de la información representada por este objeto.

      @param eventTime          Tiempo vigente del sistema gestionado.
      @param source             Objeto que emite notificación.
      @param request            Indica quién emitió la petición, el sistema de
                                gestión automática o el sistema gestionado.
      @param backupResult       Será uno de los siguientes, dependiendo del
                                éxito o fracaso de la copia de seguridad y el
                                valor de los parámetros de la operación copia
                                de seguridad:
                                - Opción de fallo – Indicación de error de
                                aplicación.
                                - Opción fuera de línea – Localización de copia
                                de seguridad fuera de línea.
                                - Opción de objeto local – El ejemplar del
                                objeto copia de seguridad.
      */

      void backupReport (
         in ExternalTimeType eventTime,
         in MONameType source,
         in RequestType request,
         in BackupResultType backupResult
      );

      /**
      La notificación de resultado de entrega es emitida por el objeto
      gestionado cuando se completa la operación de entrega.

      @param eventTime          Tiempo vigente de sistema gestionado.
      @param source             Objeto que emite la notificación.
      @param deliverId          El ID de entrega único suministrado con la
                                operación de entrega (cuando se proporciona).
      @param deliver            Los resultados de la operación de entrega. Es
                                del tipo UIDType, de modo que los resultados
                                puedan ser localizados. Los resultados de
                                entrega suministrados en esta Recomendación
                                pueden encontrarse en el módulo
                                DeliverResultConst.
      @param additionalInfo     Información específica de la aplicación.
      */

      void deliverResultNotification (
         in ExternalTimeType eventTime,
         in MONameType source,
         in DeliverIdTypeOpt deliverId,
         in DeliverResultType deliver,
         in AdditionalInformationSetType additionalInfo
      );
```

```
  /**
  La notificación de informe de restablecimiento se emite para informar el
  restablecimiento de un objeto gestionado a partir de una copia de
  seguridad previa.

  @param eventTime          Tiempo vigente del sistema gestionado.
  @param source             Objeto que emite notificación.
  @param request            Indica quién emitió la petición, el sistema de
                            gestión automática o el sistema gestionado.
  @param restoreResult      Será uno de los siguientes, dependiendo del
                            éxito o fracaso del restablecimiento y del
                            valor del atributo Future Auto Restore Source:
                            - Opción de fallo – Indicación de error de
                            aplicación.
                            - Opción fuera de línea – Localización de
                            Restablecimiento fuera de línea.
                            - Elección de objeto local – El ejemplar de
                            objeto copia de seguridad.
  */

  void restoreReport (
     in ExternalTimeType eventTime,
     in MONameType source,
     in RequestType request,
     in RestoreResultType restoreResult
  );

  /**
  La notificación de informe de validación se emite para informar la
  validación de un objeto gestionado.

  @param eventTime          Tiempo vigente del sistema gestionado.
  @param source             Objeto que emite la notificación.
  @param validateInfo       Indica el argumento dado a la operación de
                            validación.
  @param validateResult     Será uno de los siguientes, dependiendo del
                            éxito o fracaso de la operación de validación:
                            - Validación de éxito (con resultado)
                            - Validación de fallo (con resultado)
                            - Validación de éxito
                            - Validación de fallo (con error)
  */

  void validateReport (
     in ExternalTimeType eventTime,
     in MONameType source,
     in ValidateInfoType validateInfo,
     in ValidateResultType validateResult
  );
};

/**
Constantes de notificaciones
*/

const string backupReportTypeName =
  "itut_x744d1::Notifications::backupReport";
const string deliverResultNotificationTypeName =
  "itut_x744d1::Notifications::deliverResultNotification";
const string restoreReportTypeName =
  "itut_x744d1::Notifications::restoreReport";
const string validateReportTypeName =
  "itut_x744d1::Notifications::validateReport";
```

```
const string additionalInfoName = "additionalInfo";
const string backupResultName = "backupResult";
const string deliverIdName = "deliverId";
const string deliverName = "deliver";
const string eventTimeName = "eventTime";
const string sourceName = "source";
const string restoreResultName = "restoreResult";
const string requestName = "request";
const string validateInfoName = "validateInfo";
const string validateResultName = "validateResult";

/**
```

9.18 Vinculación de nombres

```
*/

module NameBinding
{
  /**
  La vinculación de nombres se utiliza para denominar el objeto
  distribuidor de soporte lógico para un objeto ManagedElement.
  */

  módulo SoftwareDistributor_ManagedElement
  {
     const string superiorClass = "itut_m3120::ManagedElement";
     const boolean superiorSubclassesAllowed = TRUE;
     const string subordinateClass =
      "itut_x744d1::SoftwareDistributor";
     const boolean subordinateSubclassesAllowed = TRUE;
     const boolean managerCreatesAllowed = TRUE;
     const DeletePolicyType deletePolicy =
         itut_x780::deleteContainedObjects;
     const string kind = "SoftwareDistributor";

  }; // módulo SoftwareDistributor_ManagedElement

  /**
  Esta vinculación de nombre se utiliza para denominar el objeto
  distribuidor de soporte lógico para un objeto subsistema.
  */

  module SoftwareDistributor_Subsystem
  {
     const string superiorClass = "itut_x780::Subsystem";
     const boolean superiorSubclassesAllowed = TRUE;
     const string subordinateClass =
         "itut_x744d1::SoftwareDistributor";
     const boolean subordinateSubclassesAllowed = TRUE;
     const boolean managerCreatesAllowed = TRUE;
     const DeletePolicyType deletePolicy =
         itut_x780::deleteContainedObjects;
     const string kind = "SoftwareDistributor";

  }; // módulo SoftwareDistributor_Subsystem

  /**
  Esta vinculación de nombre se utiliza para denominar el objeto
  distribuidor de soporte lógico para un objeto sistema.
  */
```

```
module SoftwareDistributor_System
{
   const string superiorClass = "itut_x780::System";
   const boolean superiorSubclassesAllowed = TRUE;
   const string subordinateClass =
       "itut_x744d1::SoftwareDistributor";
   const boolean subordinateSubclassesAllowed = TRUE;
   const boolean managerCreatesAllowed = TRUE;
   const DeletePolicyType deletePolicy =
       itut_x780::deleteContainedObjects;
   const string kind = "SoftwareDistributor";

}; // módulo SoftwareDistributor_System

/**
Esta vinculación de nombre se utiliza para denominar el objeto unidad de
soporte lógico para un objeto equipo.
*/

module SoftwareUnit_Equipment
{
   const string superiorClass = "itut_m3120::Equipment";
   const boolean superiorSubclassesAllowed = TRUE;
   const string subordinateClass =
       "itut_x744d1::SoftwareUnit";
   const boolean subordinateSubclassesAllowed = TRUE;
   const boolean managerCreatesAllowed = TRUE;
   const DeletePolicyType deletePolicy =
       itut_x780::deleteContainedObjects;
   const string kind = "SoftwareUnit";

}; // módulo SoftwareUnit_Equipment

/**
Esta vinculación de nombre se utiliza para denominar el objeto unidad de
soporte lógico para un objeto ManagedElement.
*/

module SoftwareUnit_ManagedElement
{
   const string superiorClass = "itut_m3120::ManagedElement";
   const boolean superiorSubclassesAllowed = TRUE;
   const string subordinateClass =
       "itut_x744d1::SoftwareUnit";
   const boolean subordinateSubclassesAllowed = TRUE;
   const boolean managerCreatesAllowed = TRUE;
   const DeletePolicyType deletePolicy =
       itut_x780::deleteContainedObjects;
   const string kind = "SoftwareUnit";

}; // módulo SoftwareUnit_ManagedElement

/**
Esta vinculación de nombre se utiliza para denominar el objeto unidad de
soporte lógico para un objeto subsistema.
*/

module SoftwareUnit_Subsystem
{
   const string superiorClass = "itut_x780::Subsystem";
   const boolean superiorSubclassesAllowed = TRUE;
   const string subordinateClass =
    "itut_x744d1::SoftwareUnit";
```

```
   const boolean subordinateSubclassesAllowed = TRUE;
   const boolean managerCreatesAllowed = TRUE;
   const DeletePolicyType deletePolicy =
    itut_x780::deleteContainedObjects;
   const string kind = "SoftwareUnit";

}; // módulo SoftwareUnit_Subsystem

/**
Esta vinculación de nombre se utiliza para denominar el objeto unidad de
soporte lógico para un objeto sistema.
*/

module SoftwareUnit_System
{
   const string superiorClass = "itut_x780::System";
   const boolean superiorSubclassesAllowed = TRUE;
   const string subordinateClass =
       "itut_x744d1::SoftwareUnit";
   const boolean subordinateSubclassesAllowed = TRUE;
   const boolean managerCreatesAllowed = TRUE;
   const DeletePolicyType deletePolicy =
       itut_x780::deleteContainedObjects;
   const string kind = "SoftwareUnit";

}; // módulo SoftwareUnit_System

/**
Esta vinculación de nombre se utiliza para denominar el objeto soporte
lógico ejecutable para un objeto equipo.
*/

module ExecutableSoftware_Equipment
{
   const string superiorClass = "itut_m3120::Equipment";
   const boolean superiorSubclassesAllowed = TRUE;
   const string subordinateClass =
       "itut_x744d1::ExecutableSoftware";
   const boolean subordinateSubclassesAllowed = TRUE;
   const boolean managerCreatesAllowed = TRUE;
   const DeletePolicyType deletePolicy =
       itut_x780::deleteContainedObjects;
   const string kind = "ExecutableSoftware";

}; // módulo ExecutableSoftware_Equipment

/**
Esta vinculación de nombre se utiliza para denominar el objeto soporte
lógico ejecutable para un objeto ManagedElement.
*/

module ExecutableSoftware_ManagedElement
{
   const string superiorClass = "itut_m3120::ManagedElement";
   const boolean superiorSubclassesAllowed = TRUE;
   const string subordinateClass =
       "itut_x744d1::ExecutableSoftware";
   const boolean subordinateSubclassesAllowed = TRUE;
   const boolean managerCreatesAllowed = TRUE;
   const DeletePolicyType deletePolicy =
       itut_x780::deleteContainedObjects;
   const string kind = "ExecutableSoftware";

}; // módulo ExecutableSoftware_ManagedElement
```

```
      /**
      Esta vinculación de nombre se utiliza para denominar el objeto soporte
      lógico ejecutable para un objeto subsistema.
      */

      module ExecutableSoftware_Subsystem
      {
         const string superiorClass = "itut_x780::Subsystem";
         const boolean superiorSubclassesAllowed = TRUE;
         const string subordinateClass =
             "itut_x744d1::ExecutableSoftware";
         const boolean subordinateSubclassesAllowed = TRUE;
         const boolean managerCreatesAllowed = TRUE;
         const DeletePolicyType deletePolicy =
             itut_x780::deleteContainedObjects;
         const string kind = "ExecutableSoftware";

      }; // módulo ExecutableSoftware_Subsystem

      /**
      Esta vinculación de nombre se utiliza para denominar el objeto soporte
      lógico ejecutable para un objeto sistema.
      */

      module ExecutableSoftware_System
      {
         const string superiorClass = "itut_x780::System";
         const boolean superiorSubclassesAllowed = TRUE;
         const string subordinateClass =
             "itut_x744d1::ExecutableSoftware";
         const boolean subordinateSubclassesAllowed = TRUE;
         const boolean managerCreatesAllowed = TRUE;
         const DeletePolicyType deletePolicy =
             itut_x780::deleteContainedObjects;
         const string kind = "ExecutableSoftware";

      }; // módulo ExecutableSoftware_System

    }; // módulo NameBinding

/**
```

9.19 Módulo FileTypeConst

```
*/

    module FileTypeConst
    {
      const string moduleName = "itut_x744d1::FileTypeConst";

      const short unstructuredText = 0; // FTAM-1
      const short unstructuredBinary = 1; // FTAM-3
      const short blockSpecial = 2;

    }; // fin del módulo FileTypeConst

/**
```

9.20 Módulo DeliverResultConst

```
*/

     module DeliverResultConst
     {
       const string moduleName = "itut_x744d1::DeliverResultConst";

       const short pass = 0;
       const short communicationError = 1;
       const short equipmentError = 2;
       const short qosError = 3;
       const short accessDenied = 4;
       const short notFound = 5;
       const short insufficientSpace = 6;
       const short alreadyDelivered = 7;
       const short inProgress = 8;
       const short unknown = 9;

     }; // fin del módulo DeliverResultConst

}; // módulo itut_x744d1

#endif //_itut_x744_1_idl_
```

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	**Redes de datos y comunicación entre sistemas abiertos**
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Impreso en Suiza
Ginebra, 2003