Unión Internacional de Telecomunicaciones

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.1142

(06/2006)

SERIE X: REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

Seguridad de las telecomunicaciones

Lenguaje de marcaje de control de acceso extensible (XACML 2.0)

Recomendación UIT-T X.1142

UIT-T

ITU Unión Internacional de Telecomunicaciones

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

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.379
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.889
Aplicaciones genéricas de la notación de sintaxis abstracta uno	X.890–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999
SEGURIDAD DE LAS TELECOMUNICACIONES	**X.1000–**

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

Recomendación UIT-T X.1142

Lenguaje de marcaje de control de acceso extensible (XACML 2.0)

Resumen

El lenguaje de marcaje de control de acceso extensible (XACML) es un tipo de lenguaje XML (lenguaje de marcaje extensible) que se utiliza para expresar políticas de control de acceso. El control de acceso consiste en decidir si se debe autorizar el acceso a recursos solicitado y hacer cumplir esa decisión. En la presente Recomendación se define el XACML básico: la sintaxis del lenguaje, los modelos y el contexto con el modelo del lenguaje de políticas, la sintaxis y las reglas de procesamiento. En la presente Recomendación se define además el perfil XACML de control de acceso basado en el cometido jerárquico y básico. Asimismo, se especifica el perfil de recursos múltiple de XACML y un perfil SAML 2.0 de XACML. Con objeto de mejorar la seguridad del intercambio de políticas basadas en XACML, en esta Recomendación se especifica también un perfil de firma digital XML XACML para la protección de los datos. Con miras a proporcionar directrices a quienes apliquen esta Recomendación, se describe un perfil de privacidad.

Desde el punto de vista técnico, la presente Recomendación es equivalente a la norma XACML 2.0 de OASIS y compatible con ella.

Orígenes

La Recomendación UIT-T X.1142 fue aprobada el 13 de junio de 2006 por la Comisión de Estudio 17 (2005-2008) del UIT-T por el procedimiento de la Recomendación UIT-T A.8.

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 en la dirección http://www.itu.int/ITU-T/ipr/.

ÍNDICE

Página

Página

Recomendación UIT-T X.1142

Lenguaje de marcaje de control de acceso extensible (XACML 2.0)

1 Alcance

La presente Recomendación define la versión 2.0 del lenguaje de marcaje de control de acceso extensible (XACML, *extensible access control markup language*), un lenguaje común para expresar una política de seguridad. El objetivo de este XACML es elaborar un lenguaje de política basado en un lenguaje de marcaje extensible (XML, *extensible markup language*) que pueda utilizarse con los siguientes fines:

- Facilitar un método para combinar determinadas reglas y políticas en un conjunto de políticas que se aplica a una petición de decisión particular.
- Facilitar un método para definir con flexibilidad el procedimiento mediante el cual se combinan reglas y políticas.
- Facilitar un método que permita tratar varios sujetos que actúan en diferentes capacidades.
- Facilitar un método para fundar una decisión de autorización en los atributos del sujeto y del recurso.
- Facilitar un método que permita abordar atributos de valores múltiples.
- Facilitar un método para fundar una decisión de autorización en los contenidos de un recurso de información.
- Facilitar un conjunto de operadores lógicos y matemáticos para los atributos del sujeto, del recurso y del entorno.
- Facilitar un método para tratar un conjunto distribuido de componentes de política, un método que permita localizar, recuperar y autenticar dichos componentes.
- Facilitar un método para identificar rápidamente la política que se aplica a una determinada acción, sobre la base de los valores de atributos de los sujetos, del recurso y de la acción.
- Facilitar una capa de abstracción que oculte los detalles del entorno de la aplicación a quienes definen las políticas.
- Facilitar un método para especificar un conjunto de acciones que deben llevarse a cabo cuando se aplique una política.

En esta Recomendación se especifican soluciones XACML para cada uno de esos requisitos. En particular, en la cláusula 7 se describe el lenguaje XACML básico: los modelos XACML, el modelo de lenguaje de política, la sintaxis de políticas y las reglas de procesamiento. En la cláusula 8 se define el perfil XACML de control de acceso basado en el cometido (RBAC, *role based access control*) básico y jerárquico. En la cláusula 9 se define el perfil de recursos múltiples XACML. En la cláusula 10 se examinan las tecnologías destinadas a proteger las comunicaciones XACML mediante la definición del perfil SAML 2.0 de XACML. En la cláusula 11 se aprovechan los aspectos básicos de la cláusula 10 para elaborar un perfil de firma digital XML para XACML. En la cláusula 12 se examina la combinación de perfiles XACML definidos en las cláusulas 7 a 11 mediante la elaboración de un perfil de recursos jerárquicos de XACML. Por último, en la cláusula 13 se examinan cuestiones relativas a la privacidad.

2 Referencias

Las siguientes Recomendaciones 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 participantes en acuerdos basados en la presente Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes. El IETF mantiene una lista de RFC, junto con aquellas que han quedado obsoletas por ulteriores RFC. El W3C mantiene una lista de las Recomendaciones más recientes y otras publicaciones.

- Recomendación UIT-T X.811 (1995) | ISO/CEI 10181-2:1996, Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de autenticación.
- Recomendación UIT-T X.812 (1995) | ISO/CEI 10181-3:1996, Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de control de acceso.

- Recomendación UIT-T X.1141 (2006), Lenguaje de marcaje de aserción de seguridad (SAML 2.0).
- IETF RFC 822 (1982), *Standard for the Format of ARPA Internet Text Messages*.
- IETF RFC 2119 (1997), *Key words for use in RFCs to Indicate Requirement Levels*.
- IETF RFC 2253 (1997), *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names*.
- IETF RFC 2256 (1997), *A Summary of the X.500 (96) User Schema for use with LDAPv3*.
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.
- IETF RFC 2732 (1999), *Format for Literal IPv6 Addresses in URL's*.
- IETF RFC 2821 (2001), *Simple Mail Transfer Protocol*.
- IETF RFC 3280 (2002), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.
- W3C Canonicalization:2002, *Exclusive XML Canonicalization Version 1.0*, W3C Recommendation, Copyright © [18 de julio de 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/xml-exc-c14n/.
- W3C Datatypes:2001, *XML Schema Part 2: Datatypes*, W3C Recommendation, Copyright © [2 de mayo de 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.
- W3C Encryption:2002, *XML Encryption Syntax and Processing*, W3C Recommendation, Copyright © [10 de diciembre de 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/.
- W3C MathML:2003, *Mathematical Markup Language (MathML), Version 2.0*, W3C Recommendation, Copyright © [21 de octubre de 2003] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2003/REC-MathML2-20031021/.
- W3C Signature:2002, *XML-Signature Syntax and Processing*, W3C Recommendation, Copyright © [12 de febrero de 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/.
- W3C XML:2004, *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, Copyright © [4 de febrero de 2004] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/REC-xml/.
- W3C XPATH:1999, *XML Path Language, Version 1.0*, W3C Recommendation, Copyright © [16 de noviembre de 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/1999/REC-xpath-19991116.
- W3C XSLT:1999, *XSL Transformations (XSLT) Version 1.0*, W3C Recommendation, Copyright © [16 de noviembre de 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/1999/REC-xslt-19991116.

NOTA – La referencia a un documento en esta Recomendación, en tanto que autónomo, no le otorga el rango de una Recomendación.

3 Definiciones

A los efectos de la presente Recomendación, se aplican las definiciones que se enumeran a continuación.

3.1 Definiciones importadas

3.1.1 Esta Recomendación utiliza el siguiente término definido en la Rec. UIT-T X.811:

a) Principio.

3.1.2 Esta Recomendación utiliza los siguientes términos definidos en la Rec. UIT-T X.812:

a) Información de control de acceso

b) Usuario.

3.1.3 Esta Recomendación utiliza los siguientes términos definidos en el glosario de servicios web W3C:

a) Namespace (espacio de nombre)

b) Esquema XML.

3.1.4 Esta Recomendación utiliza los siguientes términos definidos en IETF RFC 2828:

a) Acceso

b) Control de acceso

c) Punto de administración de políticas

d) Punto de decisión de políticas

e) Punto de cumplimiento de políticas

f) Arquitectura de seguridad

g) Política de seguridad

h) Servicio de seguridad.

3.1.5 Esta Recomendación utiliza los siguientes términos definidos en IETF RFC 2396:

a) Identificador uniforme de recursos (URI)

b) Referencia de URI.

3.1.6 Esta Recomendación utiliza el siguiente término definido en la firma XML de W3C:

a) Objeto de datos.

3.2 Definiciones adicionales

3.2.1 **acceso**: Desempeño de una acción.

3.2.2 **control de acceso**: Controlar el acceso con arreglo a una política.

3.2.3 **acción**: Operación efectuada en un recurso.

3.2.4 **política aplicable**: Las políticas y los conjuntos de políticas que rigen el acceso a una petición de decisión específica.

3.2.5 **atributo**: Característica de un sujeto, recurso, acción o entorno a la que puede hacerse referencia en un predicado u objetivo.

3.2.6 **autoridad de atributo (AA, *attribute authority*)**: Entidad que vincula atributos a identidades. Esa vinculación puede expresarse utilizando una aserción de atributo SAML con el atributo Autoridad como expedidor.

3.2.7 **decisión de autorización**: Resultado de evaluar la política aplicable, que el PDP notifica al PEP. Es una función cuyo resultado es "Permit" (permitir), "Deny" (denegar), "Indeterminate" (indeterminado) o "NotApplicable" (no aplicable) y, también puede ser (facultativo), un conjunto de obligaciones.

3.2.8 **multiconjunto**: Colección desorganizada de valores en la cual puede haber valores duplicados.

3.2.9 **condición**: Expresión de predicados. Es una función cuyo resultado puede ser "True" (verdadero), "False" (falso) o "Indeterminate" (indeterminado).

3.2.10 **secuencia conjuntiva**: Secuencia de predicados combinada mediante la operación lógica 'AND' (Y).

3.2.11 **contexto**: Representación canónica de una petición de decisión y una decisión de autorización.

3.2.12 **función de servicio de contexto**: Entidad del sistema que convierte el formato propio de las peticiones de decisión en la forma canónica de XACML, y convierte la forma canónica XACML de las decisiones de autorización en el formato propio de la respuesta.

3.2.13 **custodio**: Entidad a la que se confía cierta información personal.

3.2.14 **objeto de datos**: Se refiere a un objeto digital que se está firmando. Se hace referencia a un objeto de datos dentro de un elemento `<Reference>` utilizando un URI.

3.2.15 **decisión**: Resultado de la evaluación de una regla, una política o un conjunto de políticas.

3.2.16 **petición de decisión**: Petición realizada por un PEP a un PDP para que tome una decisión de autorización.

3.2.17 **secuencia disyuntiva**: Secuencia de predicados combinada utilizando la operación lógica 'OR' (O).

3.2.18 **efecto**: Consecuencia prevista de una regla que se cumple (ya sea "Permit" o "Deny").

3.2.19 **entorno**: Conjunto de atributos que son pertinentes a una decisión de autorización e independientes de un sujeto, un recurso o una acción particulares.

3.2.20 **política HasPrivilegesOfRole**: Tipo optativo de `<Policy>` que puede ser incluido en un `<PolicySet>` de autorización para admitir preguntas acerca de si un sujeto "tiene los privilegios de" un cometido específico.

3.2.21 **recurso jerárquico**: Recurso que se organiza como un árbol o bosque (gráfico acíclico dirigido) de determinados recursos llamados nodos.

3.2.22 **cometido subalterno**: En una jerarquía de cometidos, el cometido A es *subalterno* con respecto al cometido B si éste hereda todas las autorizaciones asociadas con el cometido A.

3.2.23 **autorizaciones para múltiples cometidos**: Conjunto de autorizaciones que el usuario sólo puede obtener si tiene varios cometidos simultáneamente.

3.2.24 **atributo denominado**: Ejemplar específico de un atributo, determinado por el nombre y tipo de atributo, la identidad del titular del atributo (que puede ser de varios tipos: sujeto, recurso, acción o entorno) y (optativamente) la identidad de la autoridad expedidora.

3.2.25 **nodo**: Determinado recurso que forma parte de un recurso jerárquico.

3.2.26 **obligación**: Operación definida en una política o conjunto de políticas que debería realizar el PEP cuando ejecuta una decisión de autorización.

3.2.27 **propietario**: Sujeto de información personal.

3.2.28 **autorización**: Capacidad o derecho para llevar a cabo alguna acción o algún recurso, posiblemente sólo bajo ciertas condiciones indicadas.

3.2.29 **conjunto de <políticas de autorización> (PPS, *permission `<policy set>`*)**: `<PolicySet>` que contiene las autorizaciones propiamente dichas asociadas con un determinado cometido.

3.2.30 **punto de información de política (PIP, *policy information point*)**: Entidad del sistema que actúa como fuente de valores de atributo.

3.2.31 **conjunto de políticas**: Una serie de políticas, otros conjuntos de políticas, un algoritmo de combinación de políticas y (optativamente) un conjunto de obligaciones. Puede constituir un componente de otro conjunto de políticas.

3.2.32 **predicado**: Afirmación sobre atributos cuya veracidad puede ser evaluada.

3.2.33 **recurso**: Componente de los datos, del servicio o del sistema.

3.2.34 **cometido**: Función laboral en el contexto de una organización, con una semántica asociada que describe la autoridad y responsabilidad conferidas al usuario asignado a ese cometido.

3.2.35 **control de acceso basado en el cometido (RBAC, *role based access control*)**: Modelo de control de acceso a los recursos en el que las acciones autorizadas con los recursos se definen según los cometidos y no según las identidades de los sujetos.

3.2.36 **autoridad habilitadora de cometidos**: Entidad que asigna atributos y valores de cometido a los usuarios o habilita atributos y valores de cometido durante una sesión de usuario.

3.2.37 **conjunto de <políticas de un cometido> (RPS, *role `<PolicySet>`*)**: `<PolicySet>` que asocia a los titulares de un determinado atributo y valor de cometido con un `<PolicySet>` de autorización que contiene las autorizaciones propiamente dichas asociadas con ese cometido.

3.2.38 **cometido superior**: En una jerarquía de cometidos, el cometido A es *superior* al cometido B si el cometido A hereda todas las autorizaciones asociadas con el cometido B.

3.2.39 **regla**: Objetivo, efecto y condición. Es uno de los componentes de una política.

3.2.40 **algoritmo de combinación de reglas**: Procedimiento para combinar decisiones a partir de numerosas reglas.

3.2.41 **sujeto**: Participante cuyos atributos se pueden describir mediante un predicado.

3.2.42 **objetivo**: El conjunto de peticiones de decisión, identificado por definiciones de recurso, sujeto y acción, que se ha de evaluar aplicando una regla, una política o un conjunto de políticas.

3.2.43 **unificación de tipos**: Método mediante el cual se "unifican" dos expresiones de tipo. Las expresiones de tipo se comparan en toda su estructura. Cuando una variable de tipo aparece en una expresión, se "unifica" entonces para representar el elemento de estructura correspondiente de la otra expresión, ya sea otra variable o una subexpresión. Todas las asignaciones de variable deben ser coherentes en ambas estructuras. No se logra la unificación si no es posible armonizar las dos expresiones, porque tienen estructuras disímiles o porque hay conflictos de instanciación, como cuando una variable necesita representar "**xs:string**" y "**xs:integer**".

4 Abreviaturas, siglas o acrónimos

A los efectos de esta Recomendación se utilizan las siguientes abreviaturas, siglas o acrónimos.

AA	Autoridad de atributos (*attribute authority*)
ASP	Proveedor de servicio de aplicación (*application service provider*)
CA	Autoridad de certificación (*certification authority*)
CMP	Protocolo de gestión de certificados (*certificate management protocol*)
CRL	Lista de revocación de certificados (*certificate revocation list*)
ECP	Cliente/mandatario mejorado (*enhanced client/proxy*)
HTTP	Protocolo de transferencia de hipertexto (*hypertext transfer protocol*)
ID	Identificador
IPSEC	Protocolo de seguridad del protocolo Internet (*IP SECurity protocol*)
LDAP	Protocolo ligero de acceso al directorio (*lightweight directory access protocol*)
PAP	Punto de administración de la política (*policy administration point*)
PDP	Punto de decisión de la política (*policy decision point*)
PEP	Punto de ejecución de la política (*policy enforcement point*)
PIP	Punto de información de política (*policy information point*)
PKI	Infraestructura de clave pública (*public-key infrastructure*)
POP	Prueba de posesión (*proof of possession*)
PPS	Conjunto de políticas de autorización (*permission* `<policy set>`)
RA	Autoridad de registro (*registration authority*)
RBAC	Control de acceso basado en el cometido (*role based access control*)
RPS	Conjunto de políticas de cometido (*role* `<policyset>`)
RSA	(Algoritmo de clave pública) Rivest Shamir Adleman (*Rivest Shamir Adleman (public key algorithm)*)
SAML	Lenguaje de marcaje de aserción de seguridad (*security assertion markup language*)
SP	Proveedor de servicio (*service provider*)
SSO	Inicio de sesión única (*single sign on*)
TLS	Seguridad de la capa de transporte (*transport layer security*)
URI	Identificador uniforme de recursos (*uniform resource identifier*)
URN	Nombre de recurso uniforme (*uniform resource name*)
XML	Lenguaje de marcaje extensible (*extensible markup language*)
XPath	Trayecto XML (*XML path*)
XSLT	Transformación del lenguaje de hojas de estilo extensible (*extensible stylesheet language transformation*)

5 Convenios

En la presente Recomendación se utilizan algunas palabras clave y formas gramaticales para indicar la cualidad de una especificación: "debe", "no debe", el verbo en futuro, afirmativo o negativo, "hay que", "debería", "no debería", "recomendado", "puede" o "podrá" y "facultativo". Todos estos términos se deben interpretar conforme a la norma RFC 2119.

En esta Recomendación, la expresión "Normativa, pero facultativa" significa que la funcionalidad descrita es facultativa en cuanto a la conformidad de un sistema con las implementaciones XACML, pero si se afirma que el sistema soporta la funcionalidad con arreglo a este perfil, es obligatorio implantarla en la forma descrita.

En las descripciones de sintaxis, los elementos entre corchetes angulares ("<", ">") se deben sustituir por valores apropiados, los corchetes ("[", "]") indican elementos facultativos, los elementos entre comillas son componentes literales y "*" indica que el elemento precedente puede aparecer varias veces o no aparecer.

6 Presentación general

Las "economías de escala" han incitado a los vendedores de plataformas informáticas a elaborar productos con una funcionalidad sumamente generalizada para que puedan utilizarse en el mayor número posible de situaciones. Tal como se comercializan, estos productos ofrecen el máximo de posibilidades para acceder a datos y ejecutar programas informáticos, de tal forma que puedan utilizarse en tantos entornos de aplicación como sea posible, incluidos aquellos que tienen las políticas de seguridad más permisivas. En el caso más común de una política de seguridad relativamente restrictiva, es preciso limitar los privilegios inherentes a la plataforma en la configuración.

La política de seguridad de una gran empresa tiene numerosos elementos y muchos puntos de ejecución. Los elementos de una política pueden estar administrados por los departamentos de sistemas informáticos. personal jurídico y finanzas. Para hacer efectiva una política se puede utilizar la extranet, el correo, las redes WAN y los sistemas de acceso a distancia, plataformas que implementan intrínsecamente una política de seguridad permisiva. La práctica habitual es administrar independientemente la configuración de cada punto de ejecución con miras a implementar la política de seguridad en la forma más rigurosa posible. Por consiguiente, modificar la política de seguridad es onerosa y poco fiable. Además, prácticamente es imposible obtener una visión global de las medidas de protección en vigor en toda la empresa para hacer cumplir la política. También es cada vez mayor la presión ejercida por los consumidores, las partes interesadas y los organismos reguladores sobre las empresas y los responsables gubernamentales para que se utilicen las "prácticas más idóneas" en la protección de los activos de información de la empresa y sus clientes.

Por estos motivos, es urgente encontrar un lenguaje común que exprese una política de seguridad. Si se implementa globalmente en una empresa, ésta, gracias a un lenguaje de política común, podrá administrar el cumplimiento de todos los elementos de su política de seguridad en todos los componentes de sus sistemas de información. La administración de una política de seguridad puede abarcar todas las etapas siguientes, o sólo algunas de ellas: redacción, revisión, prueba, aprobación, expedición, combinación, análisis, modificación, cancelación, recuperación y cumplimiento.

7 XACML básico

En esta cláusula se definen los fundamentos del XACML, que incluyen los requisitos de política generales, los modelos, el contexto general y la sintaxis de política. Se facilitan además algunos ejemplos.

7.1 Antecedentes

Esta cláusula tiene carácter informativo.

Es lógico utilizar el lenguaje XML como base de un lenguaje común de política de seguridad, debido a la facilidad con que su sintaxis y su semántica pueden ampliarse para admitir los requisitos singulares de una aplicación, y porque es muy admitido por los principales proveedores de plataformas y herramientas.

7.1.1 Requisitos

Los requisitos básicos de un lenguaje para expresar la política de seguridad de un sistema de información son los siguientes:

- Facilitar un método para combinar determinadas reglas y políticas en un conjunto de políticas que se aplica a una petición de decisión particular.
- Facilitar un método para definir con flexibilidad el procedimiento mediante el cual se combinan reglas y políticas.
- Facilitar un método que permita tratar varios sujetos que actúan en diferentes capacidades.

- Facilitar un método para fundar una decisión de autorización para atributos del sujeto y del recurso.
- Facilitar un método que permita abordar atributos de valores múltiples.
- Facilitar un método para fundar una decisión de autorización para los contenidos de un recurso de información.
- Facilitar un conjunto de operadores lógicos y matemáticos para los atributos del sujeto, del recurso y del entorno.
- Facilitar un método para tratar un conjunto distribuido de componentes de política, un método que permita localizar, recuperar y autenticar dichos componentes.
- Facilitar un método para identificar rápidamente la política que se aplica a una determina acción, sobre la base de los valores de atributos de los sujetos, del recurso y de la acción.
- Facilitar una capa de abstracción que oculte los detalles del entorno de la aplicación a quienes definen las políticas.
- Facilitar un método para especificar un conjunto de acciones que deben llevarse a cabo cuando se aplique una política.

El objetivo de este XACML es expresar estos principios reconocidos en el campo de la política de control de acceso utilizando un lenguaje que sea una extensión de XML. En las cláusulas siguientes se examinan las soluciones XACML para cada uno de dichos requisitos.

7.1.2 Combinación de reglas y políticas

La política completa aplicable a una determinada petición de decisión puede estar compuesta por varias reglas o políticas individuales. Por ejemplo, en una aplicación de privacidad personal, el propietario de la información personal puede definir ciertos aspectos de la política de divulgación, y la empresa que custodia esa información puede definir otros. A fin de tomar una decisión de autorización, debe ser posible combinar las dos políticas separadas para formar la política única aplicable a la petición.

XACML define los tres elementos de política más generales: `<Rule>`, `<Policy>` y `<PolicySet>`. El elemento `<Rule>` contiene una expresión booleana que puede evaluarse aisladamente, pero que un PDP no puede tratar aisladamente. Por lo tanto, no tiene por objeto formar la base de una decisión de autorización por sí misma. Sólo podrá existir aisladamente en un PAP de XACML, donde puede constituir la unidad básica de gestión y ser reutilizada en múltiples políticas.

El elemento `<Policy>` contiene un conjunto de elementos `<Rule>` y un procedimiento especificado para combinar los resultados de su evaluación. Es la unidad de política básica utilizada por el PDP y, por lo tanto, se considera que es la base de una decisión de autorización.

El elemento `<PolicySet>` contiene un conjunto de elementos `<Policy>` u otros `<PolicySet>` y un procedimiento especificado para combinar los resultados de su evaluación. Es el medio normalizado para combinar políticas separadas en una sola política combinada.

7.1.3 Algoritmos de combinación

XACML define varios algoritmos de combinación que pueden ser identificados, respectivamente, por un atributo `RuleCombiningAlgId` o `PolicyCombiningAlgId` de los elementos `<Policy>` o `<PolicySet>`. El algoritmo de combinación de reglas define un procedimiento para tomar una decisión de autorización en función de los distintos resultados de la evaluación de un conjunto de reglas. De modo similar, el algoritmo de combinación de políticas define un procedimiento para tomar una decisión de autorización en función de los distintos resultados de la evaluación de un conjunto de políticas. Hay algoritmos de combinación normalizados para los siguientes casos:

- denegación-prioritaria (ordenada y no ordenada),
- permiso-prioritario (ordenado y no ordenado),
- se aplica el primero y
- sólo una es aplicable.

En el caso del algoritmo "Denegación-prioritaria", si se encuentra un solo elemento `<Rule>` o `<Policy>` que produzca en la evaluación el resultado "Denegación", entonces, independientemente del resultado de la evaluación de los otros elementos `<Rule>` o `<Policy>` en la política aplicable, el resultado combinado será "Denegación".

Del mismo modo, en el caso del algoritmo "Permiso-prioritario", si se encuentra un solo resultado "Permiso", el resultado combinado será "Permiso".

En el caso del algoritmo de combinación "Se aplica el primero", el resultado combinado será el resultado de la evaluación del primer elemento <Rule>, <Policy> o <PolicySet> en la lista de reglas cuyo objetivo es aplicable a la solicitud de decisión.

El algoritmo de combinación de políticas "Sólo una es aplicable" sólo se aplica a políticas. El resultado de este algoritmo combinante garantiza que sólo es aplicable una política o un conjunto de políticas en virtud de sus objetivos. Si no hay ninguna política o conjunto de políticas aplicables, el resultado es "NotApplicable", pero si hay más de una política o conjunto de políticas aplicables, el resultado es "Indeterminado". Cuando sólo es aplicable una política o conjunto de políticas, el resultado del algoritmo combinante es el resultado de la evaluación de la única política o conjunto de políticas aplicable.

Las políticas y los conjuntos de políticas pueden tomar parámetros que modifican el comportamiento de los algoritmos combinantes. Sin embargo ninguno de los algoritmos combinantes normalizados es afectado por parámetros.

Los usuarios de esta Recomendación pueden definir sus propios algoritmos combinantes.

7.1.4 Sujetos múltiples

Muchas políticas de control de acceso están basadas en las acciones de más de un sujeto. Por ejemplo, la política que rige la ejecución de una transacción financiera de alto valor puede exigir la aprobación de varias personas que actúan en diferentes cualidades. Por lo tanto, XACML reconoce que puede haber más de un sujeto pertinente para una petición de decisión. El atributo de categoría del sujeto ("subject-category") se utiliza para diferenciar entre sujetos que actúan en diferentes cualidades. Se especifican algunos valores normalizados para este atributo y los usuarios pueden definir otros adicionales.

7.1.5 Políticas basadas en atributos de sujeto y recurso

Otro requisito común es basar una decisión de autorización sobre ciertas características del sujeto distintas de su identidad. Quizá la aplicación más común de esta opción sea el cometido del sujeto. El lenguaje XACML permite este tipo de procedimientos. Los atributos de sujetos contenidos en el contexto de petición pueden ser identificados por el elemento <SubjectAttributeDesignator>. Este elemento contiene un URN que identifica el atributo. Alternativamente, el elemento <AttributeSelector> puede contener una expresión XPath sobre el contexto de petición para identificar un valor de atributo de sujeto particular por medio de su ubicación en el contexto.

El lenguaje XACML proporciona una manera normalizada de referenciar los atributos definidos en IETF RFC 2253. Esto tiene por objetivo incitar a los implementadores a utilizar identificadores de atributos normalizados para ciertos atributos de sujeto comunes.

Otro requisito común es basar una decisión de autorización en una determinada característica del recurso y no en su identidad. El lenguaje XACML permite este tipo de procedimientos. Los atributos del recurso pueden ser identificados por el elemento <ResourceAttributeDesignator>. Este elemento contiene un URN que identifica el atributo. Alternativamente, el elemento <AttributeSelector> puede contener una expresión XPath sobre el contexto de petición para identificar un valor de atributo de recurso particular por medio de su ubicación en el contexto.

7.1.6 Atributos con múltiples valores

Las técnicas más comunes para comunicar atributos (LDAP, XPath, SAML, etc.) soportan múltiples valores por atributo. Por consiguiente, cuando un PDP de XACML consulta el valor de un atributo denominado, el resultado puede contener múltiples valores. Una colección de esos valores constituye un multiconjunto, que se distingue de un conjunto porque puede contener valores duplicados (un conjunto no). Algunas veces esta situación representa un error. Algunas veces la regla XACML se satisface si cualquiera de los valores de atributo cumple los criterios expresados en la regla.

El lenguaje XACML proporciona funciones que permiten al autor de una política expresar precisamente cómo el PDP debería tratar el caso de múltiples valores de atributo. Éstas son funciones de "orden superior".

7.1.7 Políticas basadas en el contenido de un recurso

En muchas aplicaciones es necesario basar una decisión de autorización en los datos contenidos en el recurso de información para el cual se solicita acceso. Por ejemplo, un componente común de las políticas de privacidad es que una persona debe estar autorizada a leer registros de los cuales es el sujeto. La política correspondiente debe contener una referencia al sujeto identificado en el propio recurso de información.

XACML proporciona facilidades para hacer esto cuando el recurso de información puede ser representado como documento XML. El elemento <AttributeSelector> puede contener una expresión XPath sobre el contexto de petición para identificar datos en el recurso de información que se habrá de utilizar en la evaluación de política.

7.1.8 Operadores

Las políticas de seguridad de la información se aplican a los atributos de sujetos, el recurso, la acción y el entorno para tomar una decisión de autorización. En este proceso de decisión de autorización puede ser necesario comparar o calcular atributos de muchos tipos diferentes. Por ejemplo, en una aplicación financiera puede ser necesario calcular el crédito disponible de una persona, sumando su límite de crédito al saldo de su cuenta, y posiblemente habrá que comparar el resultado con el valor de la transacción. Este tipo de situaciones crea la necesidad de efectuar operaciones aritméticas sobre atributos del sujeto (saldo de la cuenta y límite de crédito) y del recurso (valor de la transacción).

Aún más común es que una política identifique los cometidos que tienen autorización para realizar una acción particular. La operación correspondiente consiste en comprobar si hay una intersección no vacía entre el conjunto de cometidos ocupados por el sujeto y el conjunto de cometidos identificados en la política. De ahí la necesidad de realizar operaciones de conjuntos.

El lenguaje XACML comprende varias funciones incorporadas y un método para añadir funciones no normalizadas. Estas funciones podrían estar anidadas para crear expresiones arbitrariamente complejas. Esto se logra con el elemento `<Apply>`. El elemento `<Apply>` tiene un atributo XML llamado `FunctionId` que identifica la función que se ha de aplicar al contenido del elemento. Cada función normalizada se define para combinaciones específicas de tipos de datos de argumento, y también se especifica su tipo de datos que se obtienen como resultado. Por consiguiente, la coherencia del tipo de datos de la política puede comprobarse en el momento en que se escribe o analiza la política. Además, los tipos de los valores de datos presentados en el contexto de petición pueden comprobarse con respecto a los valores previstos por la política para asegurar un resultado predecible.

Además de operadores con argumentos numéricos y de conjuntos, se definen operadores para argumentos de fecha, tiempo y duración.

También se definen operadores de relación (igualdad y comparación) para diversos tipos de datos, incluidos los formulados de nombre IETF RFC 822 y X.500, cadenas, URI, etc.

También deben mencionarse los operadores que se aplican a tipos de datos booleanos, que permiten la combinación lógica de predicados en una regla. Por ejemplo, en una regla puede especificarse que el acceso se permite durante horas de trabajo Y desde un terminal situado en los locales de la empresa.

7.1.9 Distribución de políticas

En un sistema distribuido puede haber declaraciones de política escritas por varios autores y que se ejecutan en varios puntos de ejecución. Además de facilitar la compilación y combinación de componentes de política independientes, este planteamiento permite actualizar las políticas según las necesidades. Las declaraciones de política en XACML pueden distribuirse de diversas maneras, pero XACML no describe ninguna manera normativa de hacerlo. Sea cual sea el medio de distribución, se espera de los PDP confirmen si la política es aplicable a la petición de decisión que está procesando, examinando el elemento `<Target>` de la política.

Es posible anexar los elementos `<Policy>` a los recursos de información a los cuales se aplican, pero también pueden mantenerse en una o varias ubicaciones de donde se recuperan para evaluación. En esos casos, la política aplicable puede estar referenciada por un identificador o ubicador estrechamente asociado con el recurso de información.

7.1.10 Indización de políticas

Para mayor eficacia de la evaluación y facilidad de gestión, la política de seguridad global de una empresa puede expresarse como múltiples componentes de política independientes. En este caso, es necesario identificar y recuperar la declaración de política aplicable y verificar que es la correcta para la acción solicitada antes de evaluarla. Ésta es la finalidad del elemento `<Target>` en XACML.

El lenguaje permite dos procedimientos:

1) Las declaraciones de política pueden estar almacenadas en una base de datos. En este caso, el PDP debería crear una consulta de base de datos para recuperar sólo las políticas aplicables al conjunto de peticiones de decisión a las cuales se espera que responda. Además, el PDP debería evaluar el elemento `<Target>` de las declaraciones de la política o el conjunto de políticas consultadas.
2) Otra solución es almacenar en el PDP todas las políticas disponibles y para evaluar sus elementos `<Target>` en el contexto de una determinada petición de decisión, a fin de identificar las políticas y los conjuntos de políticas aplicables a esa petición.

7.1.11 Capa de abstracción

Hay muchas clases de PEP. Por ejemplo, un PEP puede formar parte de una pasarela de acceso a distancia, de un servidor web o de un usuario-agente de correo electrónico. No es realista esperar que todos los PEP de una empresa expidan actualmente, o expidan en el futuro, peticiones de decisión a un PDP en un formato común. No obstante, puede

darse el caso de que una política particular tenga que ser aplicada por múltiples PEP. No sería eficaz forzar a un redactor de políticas a escribir la misma política de varias maneras diferentes a fin de acomodar los requisitos de formato de cada PEP. De modo similar, los atributos pueden estar contenidos en varios tipos de sobre (por ejemplo, certificados de atributo X.509, aserciones de atributo SAML, etc.). Por consiguiente, es necesario tener la petición y la respuesta del PDP en XACML en un formato canónico. Este formato canónico se llama contexto XACML. Su sintaxis se define en un esquema XML.

Naturalmente, los PEP conformes a XACML pueden expedir peticiones y recibir respuestas en forma de contexto XACML, pero si el formato es diferente se necesita una etapa intermedia para convertir entre el formato petición/respuesta comprendido por el PEP y el formato de contexto XACML comprendido por el PDP.

La ventaja de este procedimiento es que se pueden escribir y analizar políticas independientemente del entorno específico en el cual se aplican.

Si el formato original de la petición/respuesta está especificado en esquema XML (por ejemplo, un PEP conforme a SAML), la transformación del formato original en contexto XACML puede estar especificada en forma de lenguaje XSLT.

De modo similar, cuando el recurso al cual se solicita acceso es un documento XML, el recurso propiamente dicho puede estar incluido en el contexto de petición, o referenciado por el mismo. Entonces, mediante la utilización de expresiones XPath en la política permite incluir en la evaluación de política los valores del recurso.

7.1.12 Acciones realizadas junto con la ejecución

En muchas aplicaciones, las políticas especifican acciones que hay que realizar en lugar de acciones que se pueden realizar o además de las mismas. XACML proporciona facilidades para especificar acciones que deben llevarse a cabo junto con la evaluación de política en el elemento `<Obligations>`. No hay definiciones normalizadas de estas acciones en la versión 2.0 de XACML. Por consiguiente, para una interpretación correcta se necesita un acuerdo bilateral entre un PAP y el PEP que aplicará sus políticas. Los PEP conformes a v2.0 de XACML denegarán siempre el acceso cuando no comprendan o no puedan satisfacer todos los elementos `<Obligations>` asociados con la política aplicable. Los elementos `<Obligations>` se devuelven al PEP para su aplicación.

7.2 Modelos XACML

Esta cláusula es informativa.

En las siguientes cláusulas se presentan los modelos de flujo de datos y del lenguaje XACML.

7.2.1 Modelo de flujo de datos

En el diagrama de flujo de datos de la figura 7-1 se indican las principales partes que participan en un dominio XACML.

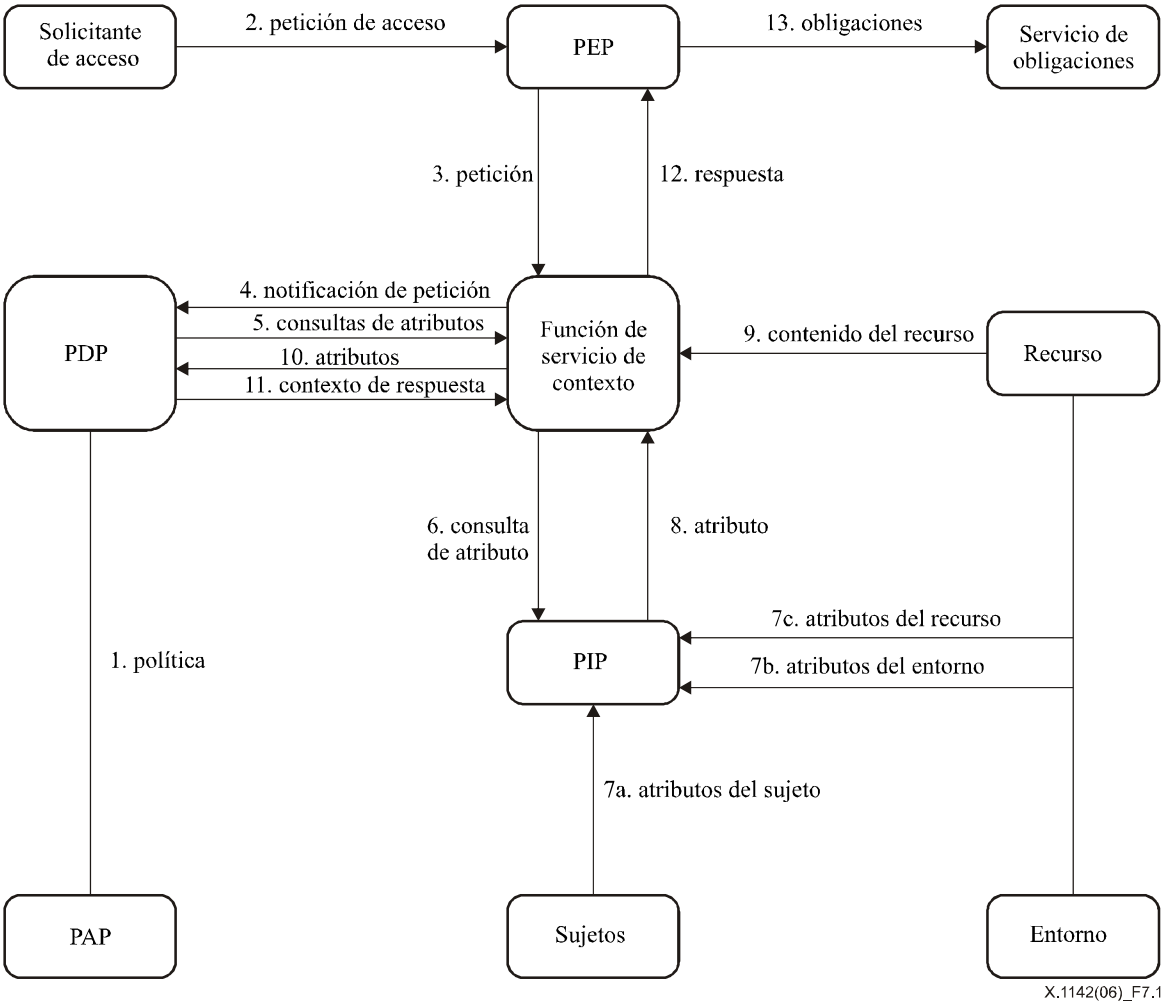

Figura 7-1/X.1142/X.1142 – Diagrama de flujo de datos

NOTA – La utilización de un depósito de datos puede facilitar algunos de los flujos del diagrama, por ejemplo las comunicaciones entre la función de servicio de contexto y el PIP, o entre el PDP y el PAP. En esta Recomendación no hay restricciones con respecto a la ubicación de estos depósitos ni se especifica un determinado protocolo de comunicación para ninguno de los flujos de datos.

En este modelo se realizan las siguientes operaciones:

1) Los PAP escriben políticas y conjuntos de políticas que se ponen a disposición del PDP. Estas políticas o conjuntos de políticas representan la política global para un determinado objetivo.
2) El solicitante de acceso envía una petición al PEP.
3) El PEP transmite la petición de acceso a la función de servicio del contexto en el formato original, posiblemente con atributos de los sujetos, el recurso, la acción y el entorno.
4) La función de servicio del contexto construye una petición XACML y la envía al PDP.
5) El PDP solicita a la función de servicio del contexto otros atributos del sujeto, el recurso, la acción y el entorno, en su caso.
6) La función de servicio del contexto solicita los atributos a un PIP.
7) El PIP obtiene los atributos solicitados.
8) El PIP devuelve los atributos solicitados a la función de servicio.
9) La función de servicio puede incluir el recurso en el contexto (es facultativo).
10) La función de servicio del contexto envía los atributos solicitados y puede enviar el recurso (es facultativo) al PDP. El PDP evalúa la política.
11) El PDP devuelve el contexto respuesta (con la decisión de autorización) a la función de servicio del contexto.
12) La función de servicio del contexto traduce el contexto respuesta al formato original del PEP y devuelve la respuesta al PEP.

13) El PEP satisface las obligaciones.

14) (No se muestra). Si se autoriza el acceso, el PEP permite acceder al recurso. Si no se obtiene autorización, deniega el acceso.

7.3 Contexto XACML

El lenguaje XACML se puede utilizar en distintos entornos de aplicaciones. El lenguaje básico está aislado del entorno de aplicación por el contexto XACML como se indica en la figura 7-2. La parte sombreada de esta figura representa el ámbito de XACML. El contexto XACML se define mediante un esquema XML que constituye la representación canónica de entradas y salidas del PDP. Los atributos señalados mediante un ejemplar de política XACML pueden presentarse como expresiones XPath sobre el contexto o como denominadores que identifican el atributo especificando el sujeto, el recurso, la acción o el entorno y su identificador, el tipo de datos y (es facultativo) el expedidor. La implementación debe convertir la representación del atributo en el entorno de aplicación (por ejemplo, SAML) y su representación en el contexto XACML. No entra en el propósito de esta Recomendación especificar cómo ha de hacerse esta conversión. En algunos casos, por ejemplo SAML, puede hacerse una conversión automática mediante una transformación XSLT (véase W3C XSLT:1999).

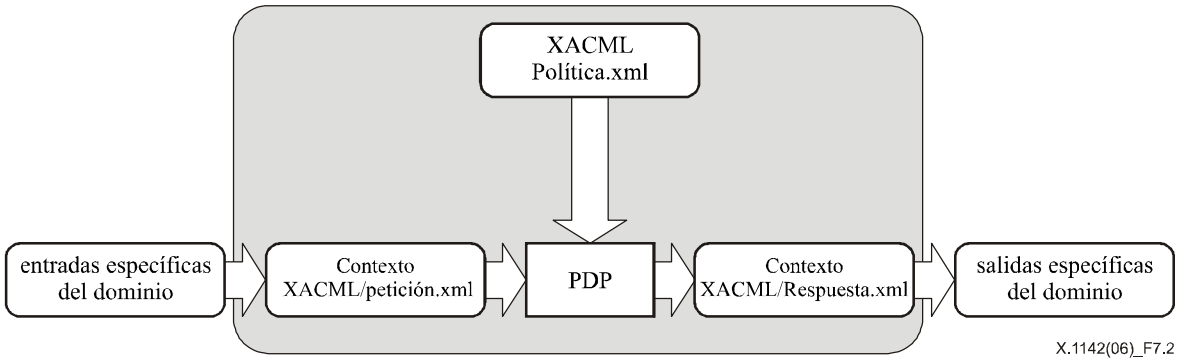

Figura 7-2/X.1142 – Contexto XACML

No es necesario que el PDP opere directamente sobre la representación XACML. Puede operar directamente sobre una representación alternativa.

7.3.1 Modelo de lenguaje de política

El modelo de lenguaje de política está representado en la figura 7-3. Los principales componentes de este modelo son:

- regla;
- política; y
- conjunto de políticas.

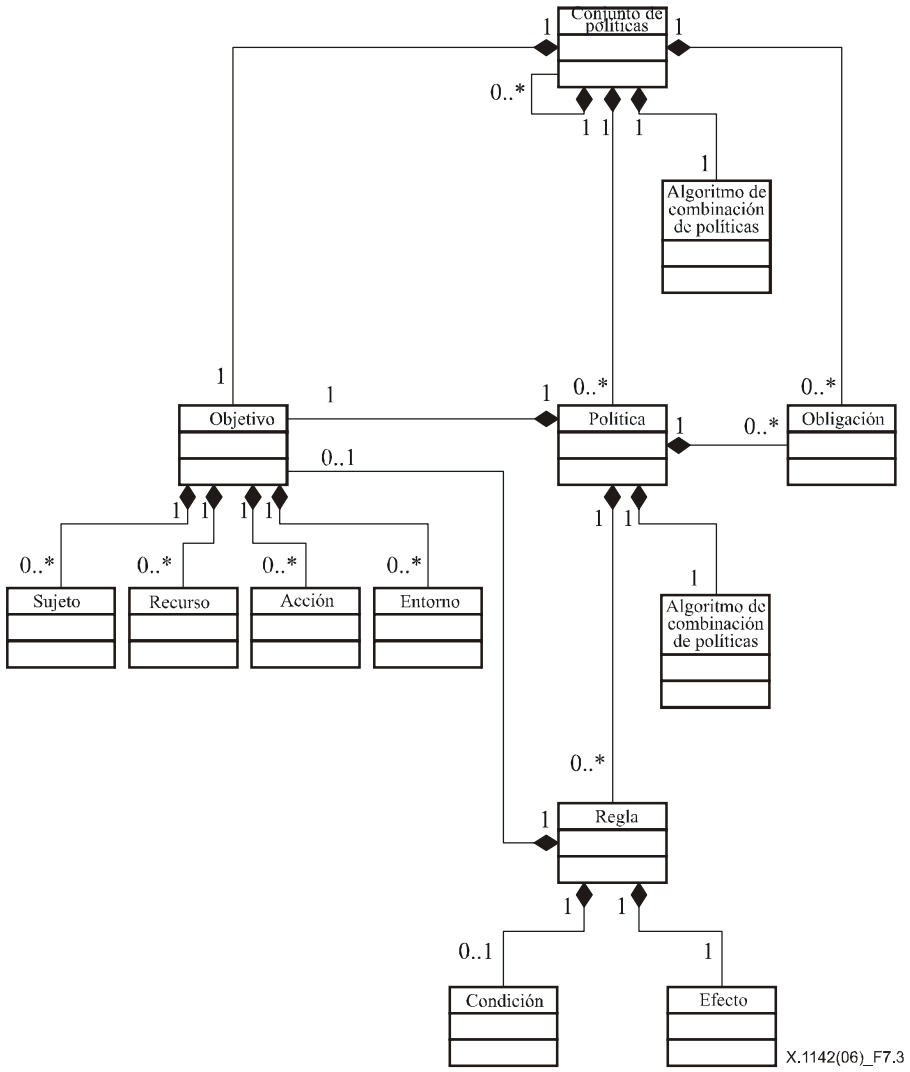

Figura 7-3/X.1142 – Modelo de lenguaje de política

7.3.1.1 Regla

La regla es la unidad elemental de la política. Una regla aislada solo es posible dentro de uno de los agentes principales del dominio XACML. Para intercambiar reglas entre los agentes principales es necesario encapsularlas en una política. La regla puede evaluarse por su contenido. Los principales componentes de una regla son:

- un objetivo;
- un efecto; y
- una condición.

7.3.1.1.1 Objetivo de una regla

El objetivo determina el conjunto de:

- los recursos;
- los sujetos;
- las acciones; y
- el entorno.

a los que se ha de aplicar la regla. El elemento `<Condition>` puede completar la especificación de aplicación del objetivo. Si la regla se ha de aplicar a todas las entidades de un determinado tipo de datos, no se indica la entidad correspondiente en el objetivo. Un PDP XACML comprueba si el objetivo corresponde a los atributos de sujeto, recurso, acción y entorno del contexto de petición. Las definiciones de los objetivos son discretas, lo que permite que el PDP identifique fácilmente las reglas aplicables.

El elemento <Target> no tiene que aparecer siempre en una <Rule>. Cuando no aparece, el objetivo de la regla es el objetivo del elemento progenitor <Policy>.

Algunos formatos de nombre de sujeto y de recursos, y algunos tipos de recursos tienen una estructura interna. Por ejemplo, el nombre de directorio X.500 y el nombre de IETF RFC 822 son nombres estructurados de sujetos, a diferencia de un nombre de cuenta, que generalmente no tiene una estructura determinada. Los nombres de caminos y URI del sistema de archivos UNIX son ejemplos de nombres estructurados. Un documento XML es un ejemplo de recurso estructurado.

Generalmente el nombre de un nodo (que no es un nodo distal) con formato estructurado también es un ejemplar válido del nombre. Por ejemplo, el nombre IETF RFC 822 "med.example.com" es un nombre IETF RFC 822 válido que identifica el conjunto de direcciones de correo que alberga el servidor med.example.com. De la misma forma, el valor XPath/XPointer //xacml-context:Request/xacml-context:Resource/xacml-context:ResourceContent/md:record/md:patient/ es un valor XPath/XPointer válido que identifica un conjunto de nodos de un documento XML.

Se plantean varias preguntas: ¿Cómo ha de interpretar el PDP un nombre que identifica una serie de sujetos o recursos, en el contexto de una política o de una petición? ¿Representan sólo el nodo identificado explícitamente por el nombre o todo el subárbol subordinado a ese nodo?

En el caso de los sujetos, dado que no hay una entidad efectiva que corresponda a estos nodos, los nombres de este tipo siempre se refieren a la serie de sujetos subordinados en la misma estructura al nodo identificado. Por tanto, los nombres de sujetos que no son distales no deberían utilizarse en las funciones de igualdad, sólo en las funciones de correspondencia, por ejemplo: "urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match", pero no con "urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal".

7.3.1.1.2 Efecto

El efecto de una regla indica la consecuencia prevista por el autor cuando el calificativo es "Verdadero" ("True"). Hay dos valores posibles: "Permit" y "Deny".

7.3.1.1.3 Condición

Es una expresión booleana que precisa las condiciones de aplicación de la regla, además de los predicados del objetivo. Es facultativo.

7.3.1.2 Política

En el modelo de flujo de datos puede verse que no hay intercambio de reglas entre entidades del sistema. Por tanto, el PAP combina las reglas en una política, que tiene cuatro componentes principales:

- un objetivo;
- un identificador de algoritmo de combinación de reglas;
- una serie de reglas; y
- obligaciones.

7.3.1.2.1 Objetivo de la política

Un elemento XACML <PolicySet>, <Policy> o <Rule> contiene un elemento <Target> que especifica el conjunto de sujetos, recursos, acciones y contextos a los que se aplica. El objetivo (<Target>) del conjunto (<PolicySet>) o de la política (<Policy>) lo puede declarar la persona que crea el <PolicySet> o la <Policy>, pero también se puede determinar a partir de los elementos <Target> de los elementos <PolicySet>, <Policy> y <Rule> que contiene.

Una entidad del sistema que determina así el <Target> no está definida por XACML, pero hay dos métodos lógicos posibles. En uno de ellos, el elemento <Target> del <PolicySet> o la <Policy> exterior ("componente exterior") será el conjunto de todos los elementos <Target> de los elementos <PolicySet>, <Policy> o <Rule> indicados ("componentes internos"). En el otro método, el elemento <Target> del componente exterior será la intersección de todos los elementos <Target> de los componentes internos. Los dos resultados serán muy diferentes: en el primer caso el elemento <Target> del componente exterior determina la aplicación a todas las peticiones de decisión que coincidan con el elemento <Target> de uno o más componentes internos; en el segundo caso el elemento <Target> del componente exterior determina la aplicación únicamente a las peticiones de decisión que coincidan con los elementos <Target> de todos los componentes internos. Se señala que el cálculo de intersección de una serie de elementos <Target> probablemente sólo puede hacerse cuando el modelo de datos del objetivo es relativamente sencillo.

Si el <Target> de una <Policy> lo ha especificado la persona que crea la política, no es indispensable incluir el elemento <Target> en los elementos <Rule> componentes de <Policy>, si es el mismo. Esos elementos <Rule> heredan el <Target> de la <Policy> a la que pertenecen.

7.3.1.2.2 Algoritmo de combinación de reglas

El algoritmo de combinación de reglas especifica los procedimientos que se utilizan para combinar los resultados de evaluación de las reglas componentes, en la evaluación de una política. Es decir, el valor de decisión que aparece en la respuesta del PDP es el valor de la política determinado por el algoritmo de combinación de reglas. Una política puede tener parámetros de combinación que modifican la aplicación del algoritmo de combinación de reglas.

7.3.1.2.3 Obligaciones

El autor de una política puede añadir obligaciones.

Cuando un PDP evalúa una política que contiene obligaciones, en la respuesta al PEP devuelve algunas de esas obligaciones.

7.3.1.3 Conjunto de políticas

Un conjunto de políticas tiene cuatro componentes principales:

- un objetivo;
- un identificador de algoritmo de combinación de políticas;
- un conjunto de políticas, y
- obligaciones.

7.3.1.3.1 Algoritmo de combinación de políticas

El algoritmo de combinación de políticas especifica los procedimientos que se utilizan para combinar los resultados de evaluación de las políticas componentes, en la evaluación de un conjunto de políticas. Es decir, el valor de Decision que aparece en la respuesta del PDP es el resultado de la evaluación del conjunto de políticas, determinado por el algoritmo de combinación de políticas. Un conjunto de políticas puede tener parámetros de combinación que modifican la aplicación del algoritmo de combinación de políticas.

7.3.1.3.2 Obligaciones

El autor de un conjunto de políticas puede añadir obligaciones, además de las obligaciones determinadas en las políticas componentes y los conjuntos de políticas.

Cuando un PDP evalúa un conjunto de políticas que contiene obligaciones, en la respuesta al PEP devuelve algunas de esas obligaciones.

7.4 Sintaxis de las políticas

Los fragmentos de código que aparecen en las siguientes cláusulas no son normativos.

7.4.1 Elemento <PolicySet>

El elemento <PolicySet> es un elemento de nivel superior en el esquema de política XACML. <PolicySet> es la suma de otros conjuntos de políticas y políticas. Es posible incluir conjuntos de políticas en un elemento global <PolicySet>, sea directamente utilizando el elemento <PolicySet> o indirectamente utilizando el elemento <PolicySetIdReference>. Las políticas pueden incluirse en un elemento global <PolicySet> directamente, utilizando el elemento <Policy>, o indirectamente con el elemento <PolicyIdReference>.

Un elemento <PolicySet> puede ser evaluado; se utilizará el procedimiento de evaluación definido en esta Recomendación.

Si un elemento <PolicySet> contiene referencias a otros conjuntos de políticas u otras políticas, en forma de URL, han de ser referencias que se puedan determinar.

Los conjuntos de políticas y las políticas incluidos en un elemento <PolicySet> se combinarán utilizando el algoritmo identificado por el atributo PolicyCombiningAlgId. <PolicySet> y <Policy> reciben el mismo tratamiento en todos los algoritmos de combinación de políticas.

El elemento <Target> determina las características de aplicación del elemento <PolicySet> a una serie de peticiones de decisión. Si el elemento <Target> incluido en el elemento <PolicySet> coincide con el contexto de la petición, el PDP podrá utilizar el elemento <PolicySet> para su decisión de autorización.

El elemento <Obligations> contiene una serie de obligaciones que se imponen al PEP para la decisión de autorización. Si el PEP no entiende una de estas obligaciones, o no puede cumplirla, actuará como si el PDP hubiera devuelto un valor de denegación de autorización ("Deny").

```
<xs:element name="PolicySet" type="xacml:PolicySetType"/>
<xs:complexType name="PolicySetType">
   <xs:sequence>
         <xs:element ref="xacml:Description" minOccurs="0"/>
         <xs:element ref="xacml:PolicySetDefaults" minOccurs="0"/>
         <xs:element ref="xacml:Target"/>
         <xs:choice minOccurs="0" maxOccurs="unbounded">
               <xs:element ref="xacml:PolicySet"/>
               <xs:element ref="xacml:Policy"/>
               <xs:element ref="xacml:PolicySetIdReference"/>
               <xs:element ref="xacml:PolicyIdReference"/>
               <xs:element ref="xacml:CombinerParameters"/>
               <xs:element ref="xacml:PolicyCombinerParameters"/>
               <xs:element ref="xacml:PolicySetCombinerParameters"/>
         </xs:choice>
         <xs:element ref="xacml:Obligations" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="PolicySetId" type="xs:anyURI" use="required"/>
   <xs:attribute name="Version" type="xacml:VersionType" default="1.0"/>
   <xs:attribute name="PolicyCombiningAlgId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

El elemento <PolicySet> es complejo, del tipo de **PolicySetType**.

El elemento <PolicySet> contiene los siguientes atributos y elementos:

– PolicySetId [Obligatorio]

  Identificador del conjunto de políticas. El PAP debe garantizar que no habrá dos políticas visibles para el PDP con el mismo identificador. Una solución es utilizar un modelo predefinido URN o URI. Si el identificador del conjunto de políticas es un URL, es condición que se pueda determinar.

– Version [Valor por defecto 1.0]

  El número de la versión del conjunto de políticas.

– PolicyCombiningAlgId [Obligatorio]

  Identificador del algoritmo de combinación de políticas que se ha de utilizar para combinar los componentes <PolicySet>, <CombinerParameters>, <PolicyCombinerParameters> y <PolicySetCombinerParameters>.

– <Description> [Facultativo]

  Descripción del conjunto de políticas en formato libre.

– <PolicySetDefaults> [Facultativo]

  Una serie de valores por defecto que se aplican al conjunto de políticas. El campo de aplicación del elemento <PolicySetDefaults> ha de ser el conjunto de políticas global.

– <Target> [Obligatorio]

  El elemento <Target> determina las características de aplicación de un conjunto de políticas a una serie de peticiones de decisión.

  El elemento <Target> puede declararlo el autor del <PolicySet>, pero también puede deducirse de los elementos <Target> de los elemento <Policy> indicados, por intersección o reunión de estos elementos.

– <PolicySet> [Cualquier número]

  Un conjunto de políticas incluido en este conjunto de políticas.

– <Policy> [Cualquier número]

  Una política incluida en este conjunto de políticas.

– <PolicySetIdReference> [Cualquier número]

  Identificador de un conjunto de políticas que se ha de incluir en este conjunto de políticas. Si <PolicySetIdReference> es un URL, es condición que se pueda determinar.

– <PolicyIdReference> [Cualquier número]

  Identificador de una política que se ha de incluir en este conjunto de políticas. Si <PolicyIdReference> es un URL, es condición que se pueda determinar.

– <Obligations> [Facultativo]

  Contiene la serie de elementos <Obligation>.

– <CombinerParameters> [Facultativo]

  Contiene una secuencia de elementos <CombinerParameter>.

– <PolicyCombinerParameters> [Facultativo]

  Contiene una secuencia de elementos <CombinerParameter> asociados a una determinada <Policy> o un elemento <PolicyIdReference> del <PolicySet>.

– <PolicySetCombinerParameters> [Facultativo]

  Contiene una secuencia de elementos <CombinerParameter> asociados a un determinado <PolicySet> o un elemento <PolicySetIdReference> del <PolicySet>.

7.4.2 Elemento <Description>

El elemento <Description> contiene una descripción en formato libre de los elementos <PolicySet>, <Policy> o <Rule>. El elemento <Description> es simple, del tipo **xs:string**.

```
    <xs:element name="Description" type="xs:string"/>
```

7.4.3 Elemento <PolicySetDefaults>

El elemento <PolicySetDefaults> especificará los valores por defecto que se aplican al elemento <PolicySet>.

```
<xs:element name="PolicySetDefaults" type="xacml:DefaultsType"/>
<xs:complexType name="DefaultsType">
   <xs:sequence>
         <xs:choice>
               <xs:element ref="xacml:XPathVersion" minOccurs="0"/>
         </xs:choice>
   </xs:sequence>
</xs:complexType>
```

El elemento <PolicySetDefaults> es complejo, del tipo **DefaultsType**.

El elemento <PolicySetDefaults> contiene el siguiente elemento:

– <XPathVersion**>** [Facultativo]

  Versión de XPath por defecto.

7.4.4 Elemento <XPathVersion>

El elemento <XPathVersion> especificará la versión de W3C XPath:1999 que han de utilizar los elementos <AttributeSelector> y las funciones de la política o el conjunto de políticas basadas en XPath.

```
<xs:element name="XPathVersion" type="xs:anyURI"/>
```

El URI para W3C XPath:1999 es "http://www.w3.org/TR/1999/Rec-xpath-19991116". El elemento <XPathVersion> es necesario si la política o el conjunto de políticas global XACML contienen elementos <AttributeSelector> o funciones basadas en XPath.

7.4.5 Elemento <Target>

El elemento <Target> identifica la serie de peticiones de decisión que van a ser evaluadas por el elemento progenitor. El elemento <Target> aparecerá como hijo de un elemento <PolicySet> y <Policy>, y también puede aparecer como hijo de un elemento <Rule>. Contiene definiciones de sujetos, recursos, acciones y entornos.

El elemento <Target> ha de contener una secuencia conjuntiva de elementos <Subjects>, <Resources> <Actions> y <Environments>. La condición de aplicación del progenitor del elemento <Target> a la petición de decisión es que haya al menos una coincidencia positiva entre cada sección del elemento <Target> y la sección correspondiente del elemento <xacml-context:Request>.

```
<xs:element name="Target" type="xacml:TargetType"/>
<xs:complexType name="TargetType">
   <xs:sequence>
          <xs:element ref="xacml:Subjects" minOccurs="0"/>
          <xs:element ref="xacml:Resources" minOccurs="0"/>
          <xs:element ref="xacml:Actions" minOccurs="0"/>
          <xs:element ref="xacml:Environments" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Target> es complejo, del tipo **TargetType**.

El elemento <Target> contiene los siguientes elementos:

– <Subjects> [Facultativo]

  Especificación de concordancia de los atributos del sujeto en el contexto. Cuando este elemento no se incluye tiene que haber concordancia del objetivo con todos los sujetos.

– <Resources> [Facultativo]

  Especificación de concordancia de los atributos del recurso en el contexto. Cuando este elemento no se incluye tiene que haber concordancia del objetivo con todos los recursos.

– <Actions> [Facultativo]

  Especificación de concordancia de los atributos de la acción en el contexto. Cuando este elemento no se incluye tiene que haber concordancia del objetivo con todas las acciones.

– <Environments> [Facultativo]

  Especificación de concordancia de los atributos del entorno en el contexto. Cuando este elemento no se incluye tiene que haber concordancia del objetivo con todos los entornos.

7.4.6 Elemento <Subjects>

El elemento <Subjects> contendrá una secuencia disyuntiva de elementos <Subject>.

```
<xs:element name="Subjects" type="xacml:SubjectsType"/>
<xs:complexType name="SubjectsType">
   <xs:sequence>
          <xs:element ref="xacml:Subject" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Subjects> es del tipo **SubjectsType** complejo.

El elemento <Subjects> contiene el siguiente elemento:

– <Subject> [Uno a muchos, Obligatorio]

  Según se define en 7.4.7.

7.4.7 Elemento <Subject>

El elemento <Subject> contendrá una secuencia conjuntiva de elementos <SubjectMatch>

```
<xs:element name="Subject" type="xacml:SubjectType"/>
<xs:complexType name="SubjectType">
   <xs:sequence>
          <xs:element ref="xacml:SubjectMatch" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Subject> es del tipo **SubjectType** complejo.

El elemento <Subject> contiene el siguiente elemento:

– <SubjectMatch> [Uno a muchos]

Una secuencia conjuntiva de correspondencias individuales de los atributos del sujeto en el contexto de petición y los valores de atributos incorporados.

7.4.8 Elemento <SubjectMatch>

El elemento <SubjectMatch> identificará a un conjunto de entidades relacionadas con el sujeto, confrontando los valores del atributo en un elemento <xacml-context:Subject> del contexto de petición con el valor del atributo incorporado.

```
<xs:element name="SubjectMatch" type="xacml:SubjectMatchType"/>
<xs:complexType name="SubjectMatchType">
   <xs:sequence>
          <xs:element ref="xacml:AttributeValue"/>
          <xs:choice>
                 <xs:element ref="xacml:SubjectAttributeDesignator"/>
                 <xs:element ref="xacml:AttributeSelector"/>
          </xs:choice>
   </xs:sequence>
   <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

El elemento <SubjectMatch> es del tipo **SubjectMatchType** complejo.

El elemento<SubjectMatch> contiene los siguientes atributos y elementos:

– MatchId [Obligatorio]

Especifica una función de correspondencia. El valor de este atributo debe ser del tipo xs:anyURI y los valores válidos se indican en 7.6.5.

– <xacml:AttributeValue> [Obligatorio]

Valor de atributo incorporado.

– <SubjectAttributeDesignator> [Definición obligatoria]

Se puede utilizar para identificar uno o más valores de atributo en un elemento <Subject> del contexto de petición.

– <AttributeSelector> [Definición obligatoria]

Se puede utilizar para identificar uno o más valores de atributo en el contexto de petición. El resultado de evaluación de la expresión XPath debería ser un atributo en un elemento <Subject> del contexto de petición.

7.4.9 Elemento <Resources>

El elemento <Resources> contendrá una secuencia disyuntiva de elementos <Resource>.

```
<xs:element name="Resources" type="xacml:ResourcesType"/>
<xs:complexType name="ResourcesType">
   <xs:sequence>
          <xs:element ref="xacml:Resource" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Resources> es del tipo **ResourcesType** complejo.

El elemento <Resources> contiene el siguiente elemento:

– <Resource> [Uno a muchos, Obligatorio]

Véase 7.4.10.

7.4.10 Elemento <Resource>

El elemento <Resource> contendrá una secuencia conjuntiva de elementos <ResourceMatch>.

```
<xs:element name="Resource" type="xacml:ResourceType"/>
<xs:complexType name="ResourceType">
   <xs:sequence>
          <xs:element ref="xacml:ResourceMatch" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Resource> es del tipo **ResourceType** complejo.

El elemento <Resource> contiene el siguiente elemento:

– <ResourceMatch> [Uno a muchos]

  Una secuencia conjuntiva de correspondencias individuales entre los atributos del recurso en el contexto de petición y los valores del atributo incorporados.

7.4.11 Elemento <ResourceMatch>

El elemento <ResourceMatch> identificará un conjunto de entidades relacionadas con el recurso, confrontando los valores del atributo en el elemento <xacml-context:Resource> del contexto de petición con el valor del atributo incorporado.

```
<xs:element name="ResourceMatch" type="xacml:ResourceMatchType"/>
<xs:complexType name="ResourceMatchType">
   <xs:sequence>
          <xs:element ref="xacml:AttributeValue"/>
          <xs:choice>
                 <xs:element ref="xacml:ResourceAttributeDesignator"/>
                 <xs:element ref="xacml:AttributeSelector"/>
          </xs:choice>
   </xs:sequence>
   <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

El elemento <ResourceMatch> es del tipo **ResourceMatchType** complejo.

El elemento <ResourceMatch> contiene los siguientes atributos y elementos:

– MatchId [Obligatorio]

  Especifica una función de correspondencia. Los valores de este atributo deben ser del tipo **xs:anyURI**, y los valores válidos se indican en 7.6.5.

– <xacml:AttributeValue> [Obligatorio]

  Valor del atributo incorporado.

– <ResourceAttributeDesignator> [Definición obligatoria]

  Puede utilizarse para identificar uno o más valores del atributo en el elemento <Resource> del contexto de petición.

– <AttributeSelector> [Definición obligatoria]

  Puede utilizarse para identificar uno o más valores de atributo en el contexto de petición. El resultado de la expresión XPath debería ser un atributo en el elemento <Resource> del contexto de petición.

7.4.12 Elemento <Actions>

El elemento <Actions> contendrá una secuencia disyuntiva de elementos <Action>.

```
<xs:element name="Actions" type="xacml:ActionsType"/>
<xs:complexType name="ActionsType">
   <xs:sequence>
          <xs:element ref="xacml:Action" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Actions> es del tipo **ActionsType** complejo.

El elemento <Actions> contiene el siguiente elemento:

– <Action> [Uno a muchos, Obligatorio]

  Véase 7.4.13.

7.4.13 Elemento <Action>

El elemento <Action> contendrá una secuencia conjuntiva de elementos <ActionMatch>.

```
<xs:element name="Action" type="xacml:ActionType"/>
<xs:complexType name="ActionType">
   <xs:sequence>
         <xs:element ref="xacml:ActionMatch" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Action> es del tipo **ActionType** complejo.

El elemento <Action> contiene el siguiente elemento:

– <ActionMatch> [Uno a muchos]

  Una secuencia conjuntiva de correspondencias individuales de los atributos de acción en el contexto de petición y los valores del atributo incorporado, véase 7.4.14.

7.4.14 Elemento <ActionMatch>

El elemento <ActionMatch> identificará un conjunto de entidades relacionadas con la acción, confrontando los valores del atributo en el elemento <xacml-context:Action> del contexto de petición con el valor del atributo incorporado.

```
<xs:element name="ActionMatch" type="xacml:ActionMatchType"/>
<xs:complexType name="ActionMatchType">
   <xs:sequence>
         <xs:element ref="xacml:AttributeValue"/>
         <xs:choice>
               <xs:element ref="xacml:ActionAttributeDesignator"/>
               <xs:element ref="xacml:AttributeSelector"/>
         </xs:choice>
   </xs:sequence>
   <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

El elemento <ActionMatch> es un tipo **ActionMatchType** complejo.

El elemento <ActionMatch> contiene los siguientes atributos y elementos:

– MatchId [Obligatorio]

  Especifica una función de correspondencia. El valor de este atributo debe ser del tipo **xs:anyURI,** y los valores válidos se indican en 7.6.5.

– <xacml:AttributeValue> [Obligatorio]

  Valor del atributo incorporado.

– <ActionAttributeDesignator> [Definición obligatoria]

  Se puede utilizar para identificar uno o más valores atributo en el elemento <Action> del contexto de petición.

– <AttributeSelector> [Definición obligatoria]

  Se puede utilizar para identificar uno o más valores de atributo en el contexto de petición. El resultado de la expresión XPath debería ser un atributo en el elemento <Action> del contexto.

7.4.15 Elemento <Environments>

El elemento <Environments> contendrá una secuencia disyuntiva de elementos <Environment>.

```
<xs:element name="Environments" type="xacml:EnvironmentsType"/>
<xs:complexType name="EnvironmentsType">
   <xs:sequence>
          <xs:element ref="xacml:Environment" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Environments> es del tipo **EnvironmentsType** complejo.

El elemento <Environments> contiene el siguiente elemento:

– <Environment> [Uno a muchos, Obligatorio]

Véase 7.4.16.

7.4.16 Elemento <Environment>

El elemento <Environment> contendrá una secuencia conjuntiva de elementos <EnvironmentMatch>.

```
<xs:element name="Environment" type="xacml:EnvironmentType"/>
<xs:complexType name="EnvironmentType">
   <xs:sequence>
          <xs:element ref="xacml:EnvironmentMatch" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Environment> es de tipo **EnvironmentType** complejo.

El elemento <Environment> contiene el siguiente elemento:

– <EnvironmentMatch> [Uno a muchos]

Una secuencia conjuntiva de correspondencias individuales entre los atributos del entorno en el contexto de petición y los valores del atributo incorporado.

7.4.17 Elemento <EnvironmentMatch>

El elemento <EnvironmentMatch> identificará un entorno, confrontando los valores de atributos en el elemento <xacml-context:Environment> del contexto de petición con el valor del atributo incorporado.

```
<xs:element name="EnvironmentMatch" type="xacml:EnvironmentMatchType"/>
<xs:complexType name="EnvironmentMatchType">
   <xs:sequence>
          <xs:element ref="xacml:AttributeValue"/>
          <xs:choice>
                 <xs:element ref="xacml:EnvironmentAttributeDesignator"/>
                 <xs:element ref="xacml:AttributeSelector"/>
          </xs:choice>
   </xs:sequence>
   <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

El elemento <EnvironmentMatch> es del tipo **EnvironmentMatchType** complejo.

El elemento <EnvironmentMatch> contiene los siguientes atributos y elementos:

– MatchId [Obligatorio]

Especifica una función de correspondencia. El valor de este atributo debe ser del tipo **xs:anyURI**, y los valores válidos se indican en 7.6.5.

– <xacml:AttributeValue> [Obligatorio]

Valor del atributo incorporado.

– <EnvironmentAttributeDesignator> [Definición obligatoria]

Se puede utilizar para identificar uno o más valores de atributos en el elemento <Environment> del contexto de petición.

– `<AttributeSelector>` [Definición obligatoria]

  Puede utilizarse para identificar uno o más valores de atributos en el contexto de petición. El resultado de la expresión XPath debería ser un atributo en el elemento `<Environment>` del contexto de petición.

7.4.18 Elemento <PolicySetIdReference>

El elemento `<PolicySetIdReference>` se utilizará para remitir a un elemento `<PolicySet>` mediante identificador. Si el elemento `<PolicySetIdReference>` es un URL, es condición que el resultado sea el elemento `<PolicySet>`. Sin embargo, el mecanismo que se utilizará para resolver una referencia a un conjunto de política está fuera del alcance de la presente Recomendación.

```
<xs:element name="PolicySetIdReference" type="xacml:IdReferenceType"/>
xs:complexType name="IdReferenceType">
   <xs:simpleContent>
          <xs:extension base="xs:anyURI">
                 <xs:attribute name="xacml:Version" type="xacml:VersionMatchType"
                 use="optional"/>
                 <xs:attribute name="xacml:EarliestVersion"
type="xacml:VersionMatchType" use="optional"/>
                 <xs:attribute name="xacml:LatestVersion"
type="xacml:VersionMatchType" use="optional"/>
          </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

El elemento `<PolicySetIdReference>` es de tipo **xacml:IdReferenceType** complejo.

IdReferenceType amplía el tipo **xs:anyURI** con los siguientes atributos:

– `Version` [Facultativo]

  Una expresión de correspondencia con una determinada versión del conjunto de políticas referenciado.

– `EarliestVersion` [Facultativo]

  Una expresión de correspondencia con una versión que es la más antigua aceptable del conjunto de políticas referenciado.

– `LatestVersion` [Facultativo]

  Una expresión de correspondencia con una versión que es la última aceptable del conjunto de políticas referenciado.

Un elemento `<PolicySetIdReference>` podría contener cualquier combinación de estos atributos. El conjunto de políticas referenciado debe corresponder a todas las expresiones. Si no figura ninguno de estos atributos, entonces es aceptable cualquier versión del conjunto de políticas. Si hubiera concordancia con varias versiones, se debería utilizar la más reciente.

7.4.19 Elemento <PolicyIdReference>

El elemento `<xacml:PolicyIdReference>` se utilizará para hacer referencia a un elemento `<Policy>` mediante identificador. Si el elemento `<PolicyIdReference>` es un URL, es condición que el resultado sea el elemento `<Policy>`. No obstante, el mecanismo que se utilizará para resolver una referencia a la política apropiada está fuera del alcance de la presente Recomendación.

```
<xs:element name="PolicyIdReference" type="xacml:IdReferenceType"/>
```

El elemento `<PolicyIdReference>` es de tipo **xacml:IdReferenceType** complejo.

7.4.20 Elemento simple tipo de versión (*Version type*)

Los elementos de este tipo contendrán el número de versión de la política o el conjunto de políticas.

```
<xs:simpleType name="VersionType">
   <xs:restriction base="xs:string">
          <xs:pattern value="(\d+\.)*\d+"/>
   </xs:restriction>
</xs:simpleType>
```

El número de la versión se expresa como una secuencia de números decimales, cada uno de ellos separado por un punto (.). 'd+' representa una secuencia de una o más cifras decimales.

7.4.21 Elemento simple de concordancia de versión (*VersionMatchType*)

Los elementos de este tipo contendrán una expresión regular restringida que corresponde a un número de versión. La expresión corresponderá a las versiones de la política o el conjunto de políticas referenciadas aceptables, que se pueden incluir en la política o el conjunto de políticas referenciantes.

```
<xs:simpleType name="VersionMatchType">
   <xs:restriction base="xs:string">
          <xs:pattern value="((\d+|\*)\.)*(\d+|\*|\+)"/>
   </xs:restriction>
</xs:simpleType>
```

La expresión de correspondencia de versión se forma separando con puntos '.', como la cadena de una versión. Un número representa una correspondencia numérica directa. El signo '*' significa que cualquier número es válido. El signo '+' significa que cualquier número es válido, al igual que cualquiera de los números subsiguientes. Así pues, las cuatro configuraciones siguientes corresponderían a la cadena de la versión '1.2.3': '1.2.3', '1.*.3', '1.2.*' y '1.+'.

7.4.22 Elemento <Policy>

El elemento `<Policy>` es la entidad más pequeña que se presentará al PDP para su evaluación.

Un elemento `<Policy>` puede someterse a evaluación, en cuyo caso se utilizará el procedimiento de evaluación definido en 7.6.10.

Los principales componentes de este elemento son los elementos `<Target>`, `<Rule>`, `<CombinerParameters>`, `<RuleCombinerParameters>` y `<Obligations>` y el atributo `RuleCombiningAlgId`.

El elemento `<Target>` define la aplicabilidad del elemento `<Policy>` a una serie de peticiones de decisión. Si el elemento `<Target>` incluido en el elemento `<Policy>` corresponde al contexto de petición, entonces el PDP podría utilizar el elemento `<Policy>` al tomar su decisión de autorización.

El elemento `<Policy>` incluye una secuencia de opciones entre los elementos `<VariableDefinition>` y `<Rule>`.

Las reglas incluidas en el elemento `<Policy>` deben combinarse mediante el algoritmo especificado por el atributo `RuleCombiningAlgId`.

El elemento `<Obligations>` contiene una serie de obligaciones que el PEP deberá cumplir al tomar la decisión de autorización.

```
<xs:element name="Policy" type="xacml:PolicyType"/>
<xs:complexType name="PolicyType">
   <xs:sequence>
          <xs:element ref="xacml:Description" minOccurs="0"/>
          <xs:element ref="xacml:PolicyDefaults" minOccurs="0"/>
          <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
          <xs:element ref="xacml:Target"/>
          <xs:choice maxOccurs="unbounded">
                 <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
                 <xs:element ref="xacml:RuleCombinerParameters" minOccurs="0"/>
                 <xs:element ref="xacml:VariableDefinition"/>
                 <xs:element ref="xacml:Rule"/>
          </xs:choice>
          <xs:element ref="xacml:Obligations" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="PolicyId" type="xs:anyURI" use="required"/>
   <xs:attribute name="Version" type="xacml:VersionType" default="1.0"/>
   <xs:attribute name="RuleCombiningAlgId" type="xs:anyURI" use="required"/>
</xs:complexType>
```

El elemento `<Policy>` es del tipo **PolicyType** complejo.

El elemento `<Policy>` 0contiene los siguientes atributos y elementos:

– `PolicyId` [Obligatorio]

  Identificador de política. El PAP debe garantizar que no hay dos políticas visibles para el PDP que tienen el mismo identificador. Esto puede lograrse siguiendo un esquema URN o URI predefinido. Si el identificador de política es un URL, es condición que entonces éste podría resolverse.

– `Version` [Por defecto 1.0]

  El número de version de la política.

– `RuleCombiningAlgId` [Obligatorio]

  El identificador del algoritmo de combinación de reglas mediante el cual deben combinarse los componentes `<Policy>`, `<CombinerParameters>` y `<RuleCombinerParameters>`.

– `<Description>` [Facultativo]

  Una descripción libre de la política

– `<PolicyDefaults>` [Facultativo]

  Una serie de valores por defecto para la política. La política que incluye las otras determinará el ámbito del elemento `<PolicyDefaults>`.

– `<CombinerParameters>` [Facultativo]

  Secuencia de parámetros que debe utilizar el algoritmo de combinación de reglas.

– `<RuleCombinerParameters>` [Facultativo]

  Secuencia de parámetros que debe utilizar el algoritmo de combinación de reglas.

– `<Target>` [Obligatorio]

  El elemento `<Target>` define la aplicabilidad de una `<Policy>` a un conjunto de peticiones de decisión.

  El elemento `<Target>` podría ser declarado por el creador del elemento `<Policy>`, o bien podría calcularse a partir de los elementos `<Target>` de los elementos `<Rule>` referenciados, por intersección o unión.

– `<VariableDefinition>` [Cualquier número]

  Definiciones de funciones comunes a las que se puede remitir desde cualquier punto de una regla donde pueda haber una expresión.

– `<Rule>` [Cualquier número]

  Una secuencia de reglas que debe combinarse de conformidad con el atributo `RuleCombiningAlgId`. Deben considerarse las reglas cuyos elementos `<Target>` concuerden con la petición de decisión. No se tendrán en cuenta los que no concuerden.

– `<Obligations>` [Facultativo]

  Secuencia conjuntiva de obligaciones que debe cumplir el PEP al tomar la decisión de autorización.

7.4.23 Elemento <PolicyDefaults>

El elemento `<PolicyDefaults>` especificará valores por defecto que se aplican al elemento `<Policy>`.

```
<xs:element name="PolicyDefaults" type="xacml:DefaultsType"/>
<xs:complexType name="DefaultsType">
   <xs:sequence>
         <xs:choice>
               <xs:element ref="xacml:XPathVersion" minOccurs="0"/>
         </xs:choice>
   </xs:sequence>
</xs:complexType>
```

El elemento `<PolicyDefaults>` es del tipo **DefaultsType** complejo.

El elemento `<PolicyDefaults>` contiene el siguiente elemento:

– `<XPathVersion>` [Facultativo]

  Versión XPath por defecto.

7.4.24 Elemento <CombinerParameters>

El elemento `<CombinerParameters>` contiene los parámetros para un algoritmo de combinación de políticas o reglas.

Si dentro de la misma política o conjunto de políticas hay múltiples elementos `<CombinerParameters>`, éstos se considerarán como un solo elemento `<CombinerParameters>` que contiene la concatenación de todas las secuencias de `<CombinerParameters>` que aparecen en todos los elementos `<CombinerParameters>` antes mencionados, conservando el orden de aparición de estos elementos en la concatenación de elementos `<CombinerParameters>`.

Cabe señalar que no se han establecido parámetros para ninguno de los algoritmos de combinación especificados en XACML 2.0.

```
<xs:element name="CombinerParameters" type="xacml:CombinerParametersType"/>
<xs:complexType name="CombinerParametersType">
   <xs:sequence>
         <xs:element ref="xacml:CombinerParameter" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <CombinerParameters> es un tipo **CombinerParametersType** complejo

El elemento <CombinerParameters> contiene el siguiente elemento:

– <CombinerParameter> [Cualquier Número]

Un único parámetro.

No es obligatorio soportar el elemento <CombinerParameters>.

7.4.25 Elemento <CombinerParameter>

El elemento <CombinerParameter> contiene un único parámetro para un algoritmo de combinación de políticas o reglas.

```
<xs:element name="CombinerParameter" type="xacml:CombinerParameterType"/>
<xs:complexType name="CombinerParameterType">
   <xs:sequence>
         <xs:element ref="xacml:AttributeValue"/>
   </xs:sequence>
   <xs:attribute name="ParameterName" type="xs:string" use="required"/>
</xs:complexType>
```

El elemento <CombinerParameter> es un tipo **CombinerParameterType** complejo.

El elemento <CombinerParameter> contiene los siguientes atributos:

– ParameterName [Obligatorio]

Identificador del parámetro

– AttributeValue [Obligatorio]

Valor del parámetro.

No es obligatorio soportar al elemento <CombinerParameter>.

7.4.26 Elemento <RuleCombinerParameters>

El elemento <RuleCombinerParameters> contiene los parámetros asociados a una regla particular dentro de una política para un algoritmo de combinación de reglas.

Cada elemento <RuleCombinerParameters> debe estar asociado con una regla de la misma política. Si hay múltiples elementos <RuleCombinerParameters> que hacen referencia a la misma regla, éstos se considerarán como un elemento <RuleCombinerParameters> que contiene la concatenación de todas las secuencias de <CombinerParameters> que hay en todos los elementos <RuleCombinerParameters> antes mencionados, conservando en la concatenación de elementos <CombinerParameter> el orden de aparición de los elementos <RuleCombinerParameters>.

Cabe señalar que no se han establecido los parámetros de ninguno de los algoritmos de combinación de reglas especificados en XACML 2.0.

```
<xs:element name="RuleCombinerParameters"
type="xacml:RuleCombinerParametersType"/>
<xs:complexType name="RuleCombinerParametersType">
   <xs:complexContent>
         <xs:extension base="xacml:CombinerParametersType">
               <xs:attribute name="RuleIdRef" type="xs:string" use="required"/>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <RuleCombinerParameters> contiene el siguiente elemento:

– RuleIdRef [Obligatorio]

Identificador de la <Rule> establecida en la política.

El elemento <RuleCombinerParameters> es facultativo únicamente cuando no se ha implementado el soporte de parámetros del combinador.

7.4.27 Elemento <PolicyCombinerParameters>

El elemento <PolicyCombinerParameters> contiene los parámetros asociados a una política particular dentro de un conjunto de políticas para un algoritmo de combinación de políticas.

Cada elemento <PolicyCombinerParameters> debe estar asociado con una política del mismo conjunto de políticas. Si hay múltiples elementos <PolicyCombinerParameters> que hacen referencia a la misma política, éstos se considerarán como un elemento <PolicyCombinerParameters> que contiene la concatenación de todas las secuencias de <CombinerParameters> que hay en todos los elementos <PolicyCombinerParameters> antes mencionados, conservando en la concatenación de elementos <CombinerParameter> el orden de aparición de los elementos <PolicyCombinerParameters>.

Cabe señalar que no se han establecido los parámetros de ninguno de los algoritmos de combinación de políticas especificados en XACML 2.0.

```
<xs:element name="PolicyCombinerParameters"
type="xacml:PolicyCombinerParametersType"/>
<xs:complexType name="PolicyCombinerParametersType">
   <xs:complexContent>
          <xs:extension base="xacml:CombinerParametersType">
                <xs:attribute name="PolicyIdRef" type="xs:anyURI"
use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <PolicyCombinerParameters> es un tipo de **PolicyCombinerParametersType** complejo.

El elemento <PolicyCombinerParameters> contiene el siguiente elemento:

– PolicyIdRef [Obligatorio]

Identificador de una <Policy> o el valor de una <PolicyIdReference> del conjunto de políticas.

El elemento <PolicyCombinerParameters> es facultativo únicamente cuando no se ha implementado el soporte de parámetros del combinador.

7.4.28 Elemento <PolicySetCombinerParameters>

El elemento <PolicySetCombinerParameters> contiene los parámetros asociados a un determinado conjunto de políticas dentro de un conjunto de políticas para un algoritmo de combinación de políticas.

Cada elemento <PolicySetCombinerParameters> debe estar asociado a un conjunto de políticas del mismo conjunto de políticas. Si hay múltiples elementos <PolicySetCombinerParameters> que hacen referencia al mismo conjunto de políticas, éstos se considerarán como un elemento <PolicySetCombinerParameters> que contiene la concatenación de todas las secuencias de <CombinerParameters>que hay en todos los elementos <PolicySetCombinerParameters> antes mencionados, conservando en la concatenación de elementos <CombinerParameter> el orden de aparición de los elementos <PolicySetCombinerParameters>.

Cabe señalar que no se han establecido los parámetros de ninguno de los algoritmos de combinación de políticas especificados en XACML 2.0.

```
<xs:element name="PolicySetCombinerParameters"
type="xacml:PolicySetCombinerParametersType"/>
<xs:complexType name="PolicySetCombinerParametersType">
   <xs:complexContent>
          <xs:extension base="xacml:CombinerParametersType">
                <xs:attribute name="PolicySetIdRef" type="xs:anyURI"
use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <PolicySetCombinerParameters> es del tipo **PolicySetCombinerParametersType** complejo.

El elemento <PolicySetCombinerParameters> contiene el siguiente elemento:

– PolicySetIdRef [Obligatorio]

Identificador de un <PolicySet> o el valor de una <PolicySetIdReference> del conjunto de políticas.

El elemento <PolicySetCombinerParameters>es facultativo únicamente cuando no se ha implementado el soporte de los parámetros del combinador.

7.4.29 Elemento <Rule>

El elemento <Rule> definirá las distintas reglas en la política. Los principales componentes de este elemento son <Target> y <Condition> y el atributo Effect.

Es posible hacer una evaluación del elemento <Rule>, en cuyo caso se recurrirá al procedimiento de evaluación descrito en 7.6.9.

```
<xs:element name="Rule" type="xacml:RuleType"/>
<xs:complexType name="RuleType">
   <xs:sequence>
         <xs:element ref="xacml:Description" minOccurs="0"/>
         <xs:element ref="xacml:Target" minOccurs="0"/>
         <xs:element ref="xacml:Condition" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="RuleId" type="xs:string" use="required"/>
   <xs:attribute name="Effect" type="xacml:EffectType" use="required"/>
</xs:complexType>
```

El elemento <Rule> es del tipo **RuleType** complejo.

El elemento <Rule> contiene los siguientes atributos y elementos:

– RuleId [Obligatorio]

Una cadena que identifica esta regla.

– Effect [Obligatorio]

Efecto de la regla. El valor de este atributo es "Permitir" o "Denegar".

– <Description> [Facultativo]

Una descripción de la regla de formato libre.

– <Target> [Facultativo]

Identifica el conjunto de peticiones de decisión que se va a evaluar aplicando el elemento <Rule>. Si este elemento se omite, entonces el objetivo de la <Rule> será definido por el elemento <Target> del elemento <Policy> circundante. Para mayores detalles véase 7.6.6.

– <Condition> [Facultativo]

Un predicado que debe satisfacerse para que se asigne a la regla su valor Effect.

7.4.30 Tipo simple de EffectType

El tipo simple de **EffectType** define los valores permitidos para el atributo Effect del elemento <Rule> y para el atributo FulfillOn del elemento <Obligation>.

```
<xs:simpleType name="EffectType">
   <xs:restriction base="xs:string">
         <xs:enumeration value="Permit"/>
         <xs:enumeration value="Deny"/>
   </xs:restriction>
</xs:simpleType>
```

7.4.31 Elemento <VariableDefinition>

El elemento <VariableDefinition> se utilizará para definir un valor que pueda ser referenciado por un elemento <VariableReference>. El nombre suministrado para este atributo VariableId no aparecerá en el atributo VariableId de ningún otro elemento <VariableDefinition> dentro de la política que lo abarca. El

elemento <VariableDefinition> puede contener un elemento <VariableReference> no definido, pero en tal caso se deberá definir más adelante el correspondiente elemento <VariableDefinition> en la política que lo abarca. Los elementos <VariableDefinition> pueden agruparse o ubicarse cerca de la referencia en la política que los abarca. Para cada elemento <VariableDefinition> puede haber varias referencias, o ninguna.

```
<xs:element name="VariableDefinition" type="xacml:VariableDefinitionType"/>
<xs:complexType name="VariableDefinitionType">
   <xs:sequence>
          <xs:element ref="xacml:Expression"/>
   </xs:sequence>
   <xs:attribute name="VariableId" type="xs:string" use="required"/>
</xs:complexType>
```

El elemento <VariableDefinition> es del tipo **VariableDefinitionType** complejo. El elemento <VariableDefinition> tiene los siguientes elementos y atributos:

– <Expression> [Obligatorio]

  Cualquier elemento de tipo complejo **ExpressionType**.

– VariableId [Obligatorio]

  Nombre de la definición de la variable.

7.4.32 Elemento <VariableReference>

El elemento <VariableReference> se utiliza para referenciar a un valor definido dentro del mismo elemento <Policy> circundante. El elemento <VariableReference> hará referencia al elemento <VariableDefinition> al ser iguales las cadenas del valor de sus respectivos atributos VariableId. El elemento <VariableReference> no puede corresponder a más de una <VariableDefinition> dentro del mismo elemento <Policy> circundante, pero podría haber cero o más elementos <VariableReference> que se refieran al mismo elemento <VariableDefinition>.

```
<xs:element name="VariableReference" type="xacml:VariableReferenceType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="VariableReferenceType">
   <xs:complexContent>
          <xs:extension base="xacml:ExpressionType">
                 <xs:attribute name="VariableId" type="xs:string" use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <VariableReference> es del tipo **VariableReferenceType** complejo, que es del tipo **ExpressionType** complejo y es miembro del grupo de sustitución del elemento <Expression>. El elemento <VariableReference> podría aparecer en cualquier lugar en el que aparezca el elemento <Expression> en el esquema.

El elemento <VariableReference> tiene el siguiente atributo:

– VariableId [Obligatorio]

  El nombre utilizado para hacer referencia al valor definido en un elemento <VariableDefinition>.

7.4.33 Elemento <Expression>

El elemento <Expression> no se utiliza directamente en una política. El elemento <Expression> indica que aparecerá en su lugar un elemento que amplía el **ExpressionType** y es miembro del grupo de sustitución del elemento <Expression>.

```
<xs:element name="Expression" type="xacml:ExpressionType" abstract="true"/>
<xs:complexType name="ExpressionType" abstract="true"/>
```

En el grupo de sustitución del elemento <Expression> se encuentran los siguientes elementos:

<Apply>, <AttributeSelector>, <AttributeValue>, <Function>, <VariableReference>, <ActionAttributeDesignator>, <ResourceAttributeDesignator>, <SubjectAttributeDesignator> y <EnvironmentAttributeDesignator>.

7.4.34 Elemento <Condition>

El elemento <Condition> es una función booleana que se aplica a los atributos o funciones de atributos del sujeto, el recurso, la acción y el entorno.

```
<xs:element name="Condition" type="xacml:ConditionType"/>
<xs:complexType name="ConditionType">
   <xs:sequence>
          <xs:element ref="xacml:Expression"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Condition> contiene un elemento <Expression>, con la restricción de que el tipo de datos que devuelve <Expression> debe ser "http://www.w3.org/2001/XMLSchema#boolean".

7.4.35 Elemento <Apply>

El elemento <Apply> indica la aplicación de una función a sus argumentos, con lo que se codifica una llamada de función. El elemento <Apply> puede aplicarse a cualquier combinación de los miembros del grupo de sustitución del elemento <Expression>.

```
<xs:element name="Apply" type="xacml:ApplyType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="ApplyType">
   <xs:complexContent>
          <xs:extension base="xacml:ExpressionType">
                 <xs:sequence>
                        <xs:element ref="xacml:Expression" minOccurs="0"
maxOccurs="unbounded"/>
                 </xs:sequence>
                 <xs:attribute name="FunctionId" type="xs:anyURI" use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <Apply> es de tipo complejo **ApplyType**.

El elemento <Apply> contiene los siguientes atributos y elementos:

– FunctionId [Obligatorio]

  El identificador de la función que se va a aplicar a los argumentos. En el anexo A se describen las funciones definidas en XACML.

– <Expression> [Facultativo]

  Argumentos para la función, que puede incluir otras funciones.

7.4.36 Elemento <Function>

El elemento <Function> se utilizará para designar una función como argumento de la función definida por el elemento progenitor <Apply>. Cuando el elemento progenitor <Apply> es una función multiconjunto de mayor orden, la función designada se aplica a todos los elementos del multiconjunto o multiconjuntos identificados en los demás argumentos del elemento progenitor.

```
<xs:element name="Function" type="xacml:FunctionType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="FunctionType">
   <xs:complexContent>
          <xs:extension base="xacml:ExpressionType">
                 <xs:attribute name="FunctionId" type="xs:anyURI" use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <Function> es de tipo complejo **FunctionType**.

El elemento <Function> contiene el siguiente atributo:

– FunctionId [Obligatorio]

  El identificador de la función.

7.4.37 Tipo complejo AttributeDesignatorType

El tipo complejo **AttributeDesignatorType** es el tipo de los elementos que identifican atributos mediante un nombre. Contiene la información requerida para la correspondencia de atributos en el contexto de petición.

También contiene información para controlar el comportamiento cuando no hay ningún atributo concordante en el contexto.

Los elementos de este tipo no modifican la semántica de correspondencia de los atributos designados, pero pueden reducir el espacio de búsqueda.

```
<xs:complexType name="AttributeDesignatorType">
   <xs:complexContent>
          <xs:extension base="xacml:ExpressionType">
                <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
                <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
                <xs:attribute name="Issuer" type="xs:string" use="optional"/>
                <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Un atributo designado concordará con un atributo si concuerda los valores de sus respectivos `AttributeId`, `DataType` y atributos `Issuer`. El `AttributeId` del designador de un atributo debe concordar (el mismo URI) con el `AttributeId` del atributo. El `DataType` del designador de un atributo debe concordar (el mismo URI) con el `DataType` del mismo atributo.

Si el atributo `Issuer` está presente en el designador de un atributo, debe corresponder efectivamente al expedidor del mismo atributo utilizando la función "`urn:oasis:names:tc:xacml:1.0:function:string-equal`". De no estar presente el `Issuer` en el designador del atributo, sólo los atributos `AttributeId` y `DataType` regirán la correspondencia del atributo con el atributo designado.

El `<AttributeDesignatorType>` contiene los siguientes atributos:

– `AttributeId` [Obligatorio]

  Este atributo especifica el `AttributeId` con el que debe concordar el atributo.

– `DataType` [Obligatorio]

  El multiconjunto que devuelve el elemento `<AttributeDesignator>` ha de contener valores de este tipo de datos.

– `Issuer` [Facultativo]

  Este atributo, si se suministra, especifica el `Issuer` con el que debe corresponder el atributo.

– `MustBePresent` [Facultativo]

  Este atributo determina si el elemento devuelve "Indeterminate" o un multiconjunto vacío en caso de que el atributo designado esté ausente del contexto de petición.

7.4.38 Elemento <SubjectAttributeDesignator>

El elemento `<SubjectAttributeDesignator>` recupera del contexto de petición un multiconjunto de valores para un atributo designado de sujeto clasificado. Un atributo del sujeto es un atributo que se incluye en un elemento `<Subject>` del contexto de petición. Un sujeto clasificado es un sujeto que se identifica mediante un atributo de una determinada categoría de sujetos. Un atributo de sujeto clasificado designado es un atributo de sujeto designado para un determinado sujeto clasificado.

El elemento `<SubjectAttributeDesignator>` devolverá un multiconjunto que contenga todos los valores de atributo del sujeto con los que concuerda el atributo de sujeto clasificado. En caso de que no haya ningún atributo concordante en el contexto, el atributo `MustBePresent` rige si este elemento devuelve un multiconjunto vacío o "Indeterminate".

El **SubjectAttributeDesignatorType** amplía la semántica de correspondencia del **AttributeDesignatorType** para reducir el espacio de búsqueda de atributos al sujeto clasificado específico, de forma que el valor del atributo `SubjectCategory` de este elemento debe concordar (el mismo URI) con el valor del atributo `SubjectCategory` del elemento `<Subject>` del contexto de petición.

Si el contexto de petición contiene múltiples sujetos con el mismo atributo XML `SubjectCategory`, éstos se tratarán como si fueran un único sujeto clasificado.

El `<SubjectAttributeDesignator>` puede aparecer en el elemento `<SubjectMatch>` y puede pasarse al elemento `<Apply>` como argumento.

```
<xs:element name="SubjectAttributeDesignator"
type="xacml:SubjectAttributeDesignatorType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="SubjectAttributeDesignatorType">
   <xs:complexContent>
         <xs:extension base="xacml:AttributeDesignatorType">
               <xs:attribute name="SubjectCategory" type="xs:anyURI"
use="optional" default="urn:oasis:names:tc:xacml:1.0:subject-category:access-
subject"/>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento `<SubjectAttributeDesignator>` es de tipo **SubjectAttributeDesignatorType**. El tipo complejo **SubjectAttributeDesignatorType** amplía el tipo complejo **AttributeDesignatorType** con un atributo `SubjectCategory`.

– `SubjectCategory` [Facultativo]

  Este atributo especifica el sujeto clasificado que es la referencia de concordancia de los atributos de sujeto designados. De no estar presente el `SubjectCategory`, se utiliza su valor por defecto "`urn:oasis:names:tc:xacml:1.0:subject-category:access-subject`".

7.4.39 Elemento <ResourceAttributeDesignator>

El elemento `<ResourceAttributeDesignator>` recupera del contexto de petición un multiconjunto de valores para un atributo de recurso designado. Un atributo del recurso es un atributo que se incluye en el elemento `<Resource>` del contexto de petición. Un atributo del recurso designado es un atributo designado que concuerda con un atributo del recurso. Se considerará que hay un atributo del recurso designado si hay al menos un atributo del recurso que se ajusta a los criterios que se establecen más adelante. Un valor de atributo del recurso es un valor que aparece en el atributo del recurso.

El elemento `<ResourceAttributeDesignator>` devolverá un multiconjunto que contenga todos los valores de atributo del recurso con los que concuerda el atributo del recurso designado. En caso de que no haya ningún atributo concordante en el contexto, el atributo `MustBePresent` determina si este elemento devuelve un multiconjunto vacío o "Indeterminate".

La concordancia de un atributo del recurso designado con un atributo del recurso estará basada en la semántica de correspondencia que se especifica en el tipo complejo **AttributeDesignatorType**.

El `<ResourceAttributeDesignator>` puede aparecer en el elemento `<ResourceMatch>` y puede pasarse al elemento `<Apply>` como argumento.

```
<xs:element name="ResourceAttributeDesignator"
type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
```

El elemento `<ResourceAttributeDesignator>` es de tipo complejo **AttributeDesignatorType**.

7.4.40 Elemento <ActionAttributeDesignator>

El elemento `<ActionAttributeDesignator>` recupera del contexto de petición un multiconjunto de valores para un atributo de acción designado. Los atributos de acción aparecen en el elemento `<Action>` del contexto de petición. Hay criterios específicos (descritos más adelante) de concordancia de un atributo de acción designado con un atributo de la acción. Se considerará que un atributo de acción designado está presente si hay al menos un atributo de acción que coincide con los criterios. Un valor de atributo de acción es un valor que aparece en el atributo de la acción.

El elemento `<ActionAttributeDesignator>` devolverá un multiconjunto con todos los valores de atributo de acción con los que concuerda el atributo de acción designada. En caso de que no haya ningún atributo concordante en el contexto, el atributo `MustBePresent` determina si este elemento devuelve un multiconjunto vacío o "Indeterminate".

La concordancia de un atributo de acción designado con un atributo de acción estará basada en la semántica de correspondencia que se especifica en el tipo complejo **AttributeDesignatorType**.

El <ActionAttributeDesignator> puede aparecer en el elemento <ActionMatch> y puede pasarse al elemento <Apply> como argumento.

```
<xs:element name="ActionAttributeDesignator" type="xacml:AttributeDesignatorType"
substitutionGroup="xacml:Expression"/>
```

El elemento <ActionAttributeDesignator> es de tipo complejo **AttributeDesignatorType**.

7.4.41 Elemento <EnvironmentAttributeDesignator>

El elemento <EnvironmentAttributeDesignator> recupera del contexto de petición un multiconjunto de valores para un atributo de entorno designado. Un atributo de entorno es un atributo que aparece en el elemento <Environment> del contexto de petición. Hay criterios específicos (descritos por debajo) de concordancia de un atributo de entorno designado con un atributo de entorno. Se considerará que un atributo de entorno designado está presente si hay al menos un atributo de entorno que coincide con los criterios. Un valor de atributo de entorno es un valor que aparece en un atributo de entorno.

Como resultado de la evaluación del elemento <EnvironmentAttributeDesignator> se obtendrá un multiconjunto con todos los valores de atributo de entorno con los que concuerda el atributo de entorno designado. En caso de que no haya ningún atributo concordante en el contexto, el atributo MustBePresent determina si este elemento devuelve un multiconjunto vacío o "Indeterminate".

La concordancia de un atributo del entorno designado con un atributo de entorno estará basada en la semántica de correspondencia que se especifica en el tipo complejo **AttributeDesignatorType**.

El <EnvironmentAttributeDesignator> puede pasarse al elemento <Apply> como argumento.

```
<xs:element name="EnvironmentAttributeDesignator" type="xacml:AttributeDesignatorType"
substitutionGroup="xacml:Expression"/>
```

El elemento <EnvironmentAttributeDesignator> es de tipo complejo **AttributeDesignatorType**.

7.4.42 Elemento <AttributeSelector>

El elemento <AttributeSelector> identifica atributos mediante su ubicación en el contexto de petición. No es obligatorio que el sistema soporte el elemento <AttributeSelector>.

El atributo XML RequestContextPath del elemento <AttributeSelector> contendrá una expresión XPath legal cuyo nodo de contexto sea el elemento <xacml-context:Request>. En la evaluación del elemento AttributeSelector se obtendrá un multiconjunto de valores del tipo especificado por el atributo DataType del elemento. Si el DataType especificado en el AttributeSelector es un tipo de dato primitivo (definido en W3C Schema:2001, W3C Datatypes:2001, 3.2), el valor léxico que devuelve la expresión XPath se convertirá en un valor del DataType especificado en el <AttributeSelector>. Si se produce un error al convertir el valor devuelto por la expresión Xpath, por ejemplo si el valor no es un ejemplar válido del DataType, el valor del <AttributeSelector> será "Indeterminate".

xs:string()

xs:boolean()

xs:integer()

xs:double()

xs:dateTime()

xs:date()

xs:time()

xs:hexBinary()

xs:base64Binary()

xs:anyURI()

Si el DataType especificado en el AttributeSelector no es uno de estos DataTypes primitivos, el AttributeSelector devolverá un multiconjunto de ejemplar del DataType especificado. Si se produce un error al convertir los valores devueltos por la expresión XPath al DataType especificado, el resultado del AttributeSelector será "Indeterminate".

Cada nodo seleccionado por la expresión XPath especificada debe ser un nodo de texto, un nodo de atributo, un nodo de instrucción de procesamiento o un nodo de comentario. La representación de cada nodo en cadena de caracteres se debe convertir en un valor de atributo del tipo de datos especificado, y el resultado del `AttributeSelector` es el multiconjunto de valores de atributo generados en todos los nodos seleccionados.

Si el nodo seleccionado por la expresión XPath especificada no es uno de los citados anteriormente (es decir, un nodo de texto, un nodo de atributo, un nodo de instrucción de procesamiento o un nodo de comentario), el resultado de la política global será "Indeterminate" y tendrá el siguiente valor `StatusCode`: "`urn:oasis:names:tc:xacml:1.0:status:syntax-error`".

```
<xs:element name="AttributeSelector" type="xacml:AttributeSelectorType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="AttributeSelectorType">
       <xs:complexContent>
           <xs:extension base="xacml:ExpressionType">
               <xs:attribute name="RequestContextPath" type="xs:string" use="required"/>
               <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
               <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
           </xs:extension>
       </xs:complexContent>
</xs:complexType>
```

El elemento `<AttributeSelector>` es de tipo complejo **AttributeSelectorType**.

El elemento `<AttributeSelector>` consta de los siguientes atributos:

– `RequestContextPath` [Obligatorio]

  Una expresión XPath cuyo nodo de contexto es el elemento `<xacml-context:Request>`. No existe ninguna restricción a la sintaxis de XPath.

– `DataType` [Obligatorio]

  El multiconjunto que devuelve el elemento `<AttributeSelector>` contendrá valores de este tipo de datos.

– `MustBePresent` [Facultativo]

  Este atributo determina si el elemento devuelve "Indeterminate" o un multiconjunto vacío en caso de que la expresión XPath no seleccione ningún nodo.

7.4.43 Elemento <AttributeValue>

El elemento `<xacml:AttributeValue>` contendrá un valor de atributo literal.

```
<xs:element name="AttributeValue" type="xacml:AttributeValueType"
substitutionGroup="xacml:Expression"/>
<xs:complexType name="AttributeValueType" mixed="true">
   <xs:complexContent>
          <xs:extension base="xacml:ExpressionType">
                 <xs:sequence>
                 <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                 </xs:sequence>
                 <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
                 <xs:anyAttribute namespace="##any" processContents="lax"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento `<xacml:AttributeValue>` es de tipo complejo **AttributeValueType**.

El elemento `<xacml:AttributeValue>` consta del siguiente atributo:

– `DataType` [Obligatorio]

  El tipo de datos del valor de atributo.

7.4.44 Elemento <Obligations>

El elemento `<Obligations>` contiene un conjunto de elementos `<Obligation>`.

No es obligatorio que el sistema soporte el elemento <Obligations>.

```
<xs:element name="Obligations" type="xacml:ObligationsType"/>
<xs:complexType name="ObligationsType">
   <xs:sequence>
          <xs:element ref="xacml:Obligation" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Obligations> es de tipo complejo **ObligationsType**.

El elemento <Obligations> contiene el siguiente elemento:

– <Obligation> [Uno a muchos]

Una secuencia de obligaciones.

7.4.45 Elemento <Obligation>

El elemento <Obligation> contendrá un identificador para la obligación y un conjunto de atributos que forman argumentos de la acción definida por la obligación. El atributo FulfillOn indica el efecto que es el motivo de esta obligación del PEP.

```
<xs:element name="Obligation" type="xacml:ObligationType"/>
<xs:complexType name="ObligationType">
   <xs:sequence>
          <xs:element ref="xacml:AttributeAssignment" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="ObligationId" type="xs:anyURI" use="required"/>
   <xs:attribute name="FulfillOn" type="xacml:EffectType" use="required"/>
</xs:complexType>
```

El elemento <Obligation> es de tipo complejo **ObligationType**. En 7.6.14 se explica cómo se determina el conjunto de obligaciones que debe devolver el PEP.

El elemento <Obligation> contiene los siguientes elementos y atributos:

– ObligationId [Obligatorio]

Identificador de obligación. El PEP interpretará el valor del identificador de obligación.

– FulfillOn [Obligatorio]

El efecto que es el motivo de esta obligación del PEP.

– <AttributeAssignment> [Facultativo]

Asignación de argumentos de obligación. El PEP interpretará los valores de los argumentos de obligación.

7.4.46 Elemento <AttributeAssignment>

El elemento <AttributeAssignment> se utiliza para incluir argumentos en las obligaciones. Contiene un AttributeId y el valor de atributo que le corresponde, mediante la ampliación de la definición del tipo **AttributeValueType**. El elemento <AttributeAssignment> se puede utilizar de cualquier forma que resulte coherente con la sintaxis del esquema, esto es, una secuencia de elementos <xs:any>. Se especificará un valor comprensible para el PEP, pero sin más precisiones en XACML.

```
<xs:element name="AttributeAssignment" type="xacml:AttributeAssignmentType"/>
<xs:complexType name="AttributeAssignmentType" mixed="true">
   <xs:complexContent>
          <xs:extension base="xacml:AttributeValueType">
                 <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
          </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <AttributeAssignment> es de tipo complejo **AttributeAssignmentType**.

El elemento <AttributeAssignment> contiene el siguiente atributo:

– AttributeId [Obligatorio]

  El identificador del atributo.

7.5 Sintaxis del contexto

Los fragmentos del esquema en las cláusulas siguientes no son normativos.

7.5.1 Elemento <Request>

El elemento <Request> es un elemento de nivel superior en el esquema del contexto XACML. El elemento <Request> es una capa de abstracción utilizada por el lenguaje de la política. Para simplificar, esta Recomendación describe la evaluación de políticas en términos de operaciones sobre el contexto. No es obligatorio que un PDP conforme represente realmente un ejemplar del contexto en forma de documento XML, pero los sistemas conformes a XACML deben producir exactamente las mismas decisiones de autorización que en el caso de transformación de todas las entradas en elementos <xacml-context:Request>.

El elemento <Request> contiene elementos <Subject>, <Resource>, <Action> y <Environment>. Puede haber múltiples elementos <Subject> y, en algunas circunstancias, múltiples elementos <Resource>. Cada elemento derivado contiene una secuencia de elementos <xacml-context:Attribute> asociados respectivamente con el sujeto, el recurso, la acción y el entorno. Estos elementos <Attribute> pueden formar parte de la evaluación de políticas.

```
<xs:element name="Request" type="xacml-context:RequestType"/>
<xs:complexType name="RequestType">
   <xs:sequence>
          <xs:element ref="xacml-context:Subject" maxOccurs="unbounded"/>
          <xs:element ref="xacml-context:Resource" maxOccurs="unbounded"/>
          <xs:element ref="xacml-context:Action"/>
          <xs:element ref="xacml-context:Environment"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Request> es de tipo complejo **RequestType**.

El elemento <Request> contiene los siguientes elementos:

– <Subject> [Uno a muchos]

  Especifica información sobre un sujeto del contexto de petición mediante la realización de un listado secuencial de elementos <Attribute> asociados con el sujeto. Se permiten uno o más elementos <Subject>. Un sujeto es una entidad asociada con la petición de acceso. Por ejemplo, un sujeto podría ser el usuario humano que inició la aplicación desde la que se emitió la petición; otro sujeto podría ser el código ejecutable de la aplicación que debe crear la petición, la máquina en la que se ejecuta la aplicación o la entidad destinataria del recurso. Los atributos de cada una de esas entidades se deben contener en elementos <Subject> distintos.

– <Resource> [Uno a muchos]

  Especifica información sobre el recurso o recursos para los que se pide acceso mediante la realización de un listado secuencial de elementos <Attribute> asociados con el recurso. Puede incluir un elemento <ResourceContent>.

– <Action> [Obligatorio]

  Especifica la acción solicitada que se va a ejecutar sobre el recurso mediante la realización de un listado de un conjunto de elementos <Attribute> asociados con la acción.

– <Environment> [Obligatorio]

  Contiene un conjunto de elementos <Attribute> para el entorno.

7.5.2 Elemento <Subject>

El elemento <Subject> especifica un sujeto mediante la realización de un listado secuencial de elementos <Attribute> asociados con el sujeto.

```
<xs:element name="Subject" type="xacml-context:SubjectType"/>
<xs:complexType name="SubjectType">
   <xs:sequence>
         <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="SubjectCategory" type="xs:anyURI"
default="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"/>
</xs:complexType>
```

El elemento <Subject> es de tipo complejo **SubjectType**.

El elemento <Subject> contiene los siguientes elementos y atributos:

– SubjectCategory [Facultativo]

  Este atributo indica el papel que desempeñó el <Subject> progenitor en la formación de la petición de acceso. Si este atributo no está presente en un elemento <Subject> dado, se utilizará el valor por defecto de "urn:oasis:names:tc:xacml:1.0:subject-category:access-subject", indicando que el elemento <Subject> progenitor representa la entidad responsable en última instancia de iniciar la petición de acceso.

  Si hay más de un elemento <Subject> que contiene un atributo "urn:oasis:names:tc:xacml:2.0:subject-category" con el mismo valor, el PDP tratará el contenido de dichos elementos como si estuvieran contenidos en el mismo elemento <Subject>.

– <Attribute> [Cualquier número]

  Una secuencia de atributos que se aplican al sujeto.

Normalmente, un elemento <Subject> contiene un <Attribute> con el siguiente AttributeId que indica la identidad del sujeto: "urn:oasis:names:tc:xacml:1.0:subject:subject-id".

Un elemento <Subject> puede contener elementos <Attribute> adicionales.

7.5.3 Elemento <Resource>

El elemento <Resource> especifica información sobre el recurso para el que se solicita el acceso, mediante la realización de un listado secuencial de elementos <Attribute> asociados con el recurso. Puede incluir el contenido del recurso.

```
<xs:element name="Resource" type="xacml-context:ResourceType"/>
<xs:complexType name="ResourceType">
   <xs:sequence>
         <xs:element ref="xacml-context:ResourceContent" minOccurs="0"/>
         <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Resource> es de tipo complejo **ResourceType**.

El elemento <Resource> contiene los siguientes elementos:

– <ResourceContent> [Facultativo]

  El contenido del recurso.

– <Attribute> [Cualquier número]

  Una secuencia de atributos del recurso.

El elemento <Resource> puede contener uno o más elementos <Attribute> con el siguiente AttributeId: "urn:oasis:names:tc:xacml:2.0:resource:resource-id". Cada uno de esos <Attribute> será una representación plena y completamente determinada de la identidad del recurso al que se quiere acceder. Si hubiera más de una de esas representaciones plenas y completamente determinadas, y se especifica algún <Attribute> con ese AttributeId se determinará un <Attribute> para cada una de las distintas representaciones de la identidad del recurso. Todos esos elementos <Attribute> deben hacer referencia al mismo ejemplar de recurso. El perfil para un recurso particular puede especificar una representación normativa única para distintos ejemplares del recurso; en este caso, cualquier <Attribute> con ese AttributeId utilizará solamente esa representación particular.

Un elemento <Resource> puede contener elementos <Attribute> adicionales.

7.5.4 Elemento <ResourceContent>

El elemento <ResourceContent> es un marcador conceptual para el contenido del recurso. Si una política XACML hace referencia al contenido del recurso mediante un elemento <AttributeSelector>, el elemento <ResourceContent> se deberá incluir en la cadena RequestContextPath.

```
<xs:complexType name="ResourceContentType" mixed="true">
    <xs:sequence>
        <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

El elemento <ResourceContent> es de tipo complejo **ResourceContentType**.

El elemento <ResourceContent> permite elementos y atributos arbitrarios.

7.5.5 Elemento <Action>

El elemento <Action> indica la acción solicitada sobre el recurso, mediante la realización de un listado de un conjunto de elementos <Attribute> asociados a la acción.

```
<xs:element name="Action" type="xacml-context:ActionType"/>
<xs:complexType name="ActionType">
    <xs:sequence>
        <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

El elemento <Action> es de tipo complejo **ActionType**.

El elemento <Action> contiene el siguiente elemento:

– <Attribute> [Cualquier número]

  Lista de atributos de la acción que se va a ejecutar sobre el recurso.

7.5.6 Elemento <Environment>

El elemento <Environment> contiene un conjunto de atributos del entorno.

```
xs:element name="Environment" type="xacml-context:EnvironmentType"/>
<xs:complexType name="EnvironmentType">
   <xs:sequence>
          <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Environment> es de tipo complejo **EnvironmentType**.

El elemento <Environment> contiene el siguiente elemento:

– <Attribute> [Cualquier número]

  Una lista de atributos del entorno. Los atributos del entorno son atributos que no están asociados ni con el recurso, ni con la acción ni con ninguno de los sujetos de la petición de acceso.

7.5.7 Elemento <Attribute>

El elemento <Attribute> es la abstracción central del contexto de petición. Contiene metadatos de atributo y uno o más valores de atributo. Los metadatos de atributo comprenden el identificador de atributos y el expedidor de atributos. Los elementos <AttributeDesignator> y <AttributeSelector> en la política pueden utilizar esos metadatos para hacer referencia a atributos.

```
<xs:element name="Attribute" type="xacml-context:AttributeType"/>
<xs:complexType name="AttributeType">
   <xs:sequence>
          <xs:element ref="xacml-context:AttributeValue" maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
   <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
   <xs:attribute name="Issuer" type="xs:string" use="optional"/>
</xs:complexType>
```

El elemento `<Attribute>` es de tipo complejo **AttributeType**.

El elemento `<Attribute>` contiene los siguientes atributos y elementos:

– `AttributeId` [Obligatorio]

  El identificador del atributo. XACML reserva varios identificadores para señalar atributos utilizados habitualmente.

– `DataType` [Obligatorio]

  El tipo de datos del contenido del elemento `<xacml-context:AttributeValue>`. Se utilizará un tipo primitivo definido en la presente Recomendación o un tipo (primitivo o estructurado) definido en un espacio de nombres declarado en el elemento `<xacml-context>`.

– `Issuer` [Facultativo]

  El expedidor de atributos. Por ejemplo, el valor de este atributo puede ser un nombre `x500Name` que se vincula a una clave pública, o algún otro identificador intercambiado en un tren de datos diferente entre el expedidor y la parte confiante.

– `<xacml-context:AttributeValue>` [Uno a muchos]

  Uno o más valores de atributo. Cada valor de atributo puede tener un contenido vacío, y puede aparecer una vez o varias veces.

7.5.8 Elemento <AttributeValue>

El elemento `<xacml-context:AttributeValue>` contiene el valor de un atributo.

```
<xs:element name="AttributeValue" type="xacml-context:AttributeValueType"/>
<xs:complexType name="AttributeValueType" mixed="true">
   <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

El elemento `<xacml-context:AttributeValue>` es de tipo complejo **AttributeValueType**.

El tipo de datos del elemento `<xacml-context:AttributeValue>` se especificará utilizando el atributo `DataType` del elemento progenitor `<Attribute>`.

7.5.9 Elemento <Response>

El elemento `<Response>` es un elemento de alto nivel en el esquema de contexto XACML. El elemento `<Response>` es una capa de abstracción utilizada por el lenguaje de la política. Cualquier sistema propietario que utilice XACML debe transformar un elemento `<Response>` de contexto XACML en el formato de su decisión de autorización.

El elemento `<Response>` encapsula la decisión de autorización producida por el PDP. Incluye una secuencia de uno o más resultados, con un elemento `<Result>` por cada recurso solicitado. Algunas implementaciones pueden devolver múltiples resultados, en particular las que soportan el perfil XACML para peticiones de múltiples recursos. No es obligatorio que el sistema soporte múltiples resultados.

```
<xs:element name="Response" type="xacml-context:ResponseType"/>
<xs:complexType name="ResponseType">
   <xs:sequence>
          <xs:element ref="xacml-context:Result" maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <Response> es de tipo complejo **ResponseType**.

El elemento <Response> contiene el siguiente elemento:

– <Result> [Uno a muchos]

  Un resultado de decisión de autorización.

7.5.10 Elemento <Result>

El elemento <Result> representa un resultado de decisión de autorización para el recurso especificado por el atributo ResourceId. Puede incluir un conjunto de obligaciones que debe cumplir el PEP. Si el PEP no comprende o no puede cumplir una obligación, debe actuar como si el PDP hubiera denegado el acceso al recurso solicitado.

```
<xs:complexType name="ResultType">
   <xs:sequence>
         <xs:element ref="xacml-context:Decision"/>
         <xs:element ref="xacml-context:Status" minOccurs="0"/>
         <xs:element ref="xacml:Obligations" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="ResourceId" type="xs:string" use="optional"/>
</xs:complexType>
```

El elemento <Result> es de tipo complejo **ResultType**.

El elemento <Result> contiene los siguientes atributos y elementos:

– ResourceId [Facultativo]

  El identificador del recurso solicitado. Si se omite este atributo, la identidad del recurso es la especificada por el atributo del recurso "urn:oasis:names:tc:xacml:1.0:resource:resource-id" en el correspondiente elemento <Request>.

– <Decision> [Obligatorio]

  La decisión de autorización: "Permit", "Deny", "Indeterminate" o "NotApplicable".

– <Status> [Facultativo]

  Indica si se han producido errores durante la evaluación de la petición de decisión y, facultativamente, proporciona información sobre dichos errores. Si el elemento <Response> contiene elementos <Result> cuyos elementos <Status> son todos idénticos, y el elemento <Response> está contenido en un contenedor de protocolo que puede transmitir información sobre el estado, esa información de estado común se podrá colocar en el contenedor de protocolo y este elemento <Status> se podrá omitir en todos los elementos <Result>.

– <Obligations> [Facultativo]

  Una lista de obligaciones que debe cumplir el PEP. Si el PEP no comprende o no puede cumplir una obligación, debe actuar como si el PDP hubiera denegado el acceso al recurso solicitado.

7.5.11 Elemento <Decision>

El elemento <Decision> contiene el resultado de la evaluación de la política.

```
<xs:element name="Decision" type="xacml-context:DecisionType"/>
<xs:simpleType name="DecisionType">
   <xs:restriction base="xs:string">
         <xs:enumeration value="Permit"/>
         <xs:enumeration value="Deny"/>
         <xs:enumeration value="Indeterminate"/>
         <xs:enumeration value="NotApplicable"/>
   </xs:restriction>
</xs:simpleType>
```

El elemento <Decision> es de tipo simple **DecisionType**.

Los valores del elemento <Decision> tienen los siguientes significados:

- Permiso: Se permite el acceso solicitado.
- Denegación: Se deniega el acceso solicitado.

– Indeterminado: El PDP es incapaz de evaluar el acceso solicitado. Los motivos de esa incapacidad son, entre otros, que faltan atributos, se producen errores al recuperar políticas, se divide por cero durante la evaluación de la política, se cometen errores de sintaxis en la petición de decisión o en la política, etc.
– No aplicable: El PDP no tiene ninguna política que se aplique a esa petición de decisión.

7.5.12 Elemento <Status>

El elemento `<Status>` representa el estado del resultado de la decisión de autorización.

```
<xs:element name="Status" type="xacml-context:StatusType"/>
<xs:complexType name="StatusType">
   <xs:sequence>
          <xs:element ref="xacml-context:StatusCode"/>
          <xs:element ref="xacml-context:StatusMessage" minOccurs="0"/>
          <xs:element ref="xacml-context:StatusDetail" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

El elemento `<Status>` es de tipo complejo **StatusType**.

El elemento `<Status>` contiene los siguientes elementos:

– `<StatusCode>` [Obligatorio]

Código de estado.

– `<StatusMessage>` [Facultativo]

Un mensaje de estado que describe el código de estado.

– `<StatusDetail>` [Facultativo]

Información adicional sobre el estado.

7.5.13 Elemento <StatusCode>

El elemento `<StatusCode>` contiene un valor principal del código de estado y una secuencia facultativa de códigos de estado secundarios.

```
<xs:element name="StatusCode" type="xacml-context:StatusCodeType"/>
<xs:complexType name="StatusCodeType">
   <xs:sequence>
          <xs:element ref="xacml-context:StatusCode" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="Value" type="xs:anyURI" use="required"/>
</xs:complexType>
```

El elemento `<StatusCode>` es de tipo complejo **StatusCodeType**.

El elemento `<StatusCode>` contiene los siguientes atributos y elementos:

– `Value` [Obligatorio]

Véase la lista de valores de B.8.

– `<StatusCode>` [Cualquier número]

Código de estado secundario. Este código de estado califica el correspondiente código de estado de orden superior.

7.5.14 Elemento <StatusMessage>

El elemento `<StatusMessage>` es una descripción libre del código de estado.

```
<xs:element name="StatusMessage" type="xs:string"/>
```

El elemento `<StatusMessage>` es de tipo **xs:string**.

7.5.15 Elemento <StatusDetail>

El elemento `<StatusDetail>` ofrece información adicional que califica al elemento `<Status>`.

```
<xs:element name="StatusDetail" type="xacml-context:StatusDetailType"/>
<xs:complexType name="StatusDetailType">
   <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

El elemento <StatusDetail> es de tipo complejo **StatusDetailType**.

El elemento <StatusDetail> permite un contenido XML arbitrario.

La inclusión de un elemento <StatusDetail> es facultativa. Sin embargo, si un PDP devuelve uno de los siguientes valores <StatusCode> definidos en XACML e incluye un elemento <StatusDetail>, se aplican las siguientes reglas.

```
urn:oasis:names:tc:xacml:1.0:status:ok
```

Un PDP no debe devolver un elemento <StatusDetail> cuando el valor de estado sea "ok".

```
urn:oasis:names:tc:xacml:1.0:status:missing-attribute
```

Un PDP puede decidir no devolver ninguna información <StatusDetail> o bien devolver un elemento <StatusDetail> que contenga uno o más elementos <xacml-context: MissingAttributeDetail>.

```
urn:oasis:names:tc:xacml:1.0:status:syntax-error
```

Un PDP no debe devolver un elemento <StatusDetail> cuando el valor de estado sea "syntax-error" (error de sintaxis). Un error de sintaxis puede deberse a un problema con la política que se esté utilizando o con el contexto de petición. El PDP puede devolver un <StatusMessage> que describa el problema.

```
urn:oasis:names:tc:xacml:1.0:status:processing-error
```

Un PDP no debe devolver un elemento <StatusDetail> cuando el valor de estado sea "processing-error" (error de procesamiento). Este código de estado señala un problema interno en el PDP. Por motivos de seguridad, el PDP puede decidir no devolver más información al PEP. En caso de error de división por cero u otro error de cálculo, el PDP puede devolver un <StatusMessage> que describa la naturaleza del error.

7.5.16 Elemento <MissingAttributeDetail>

El elemento <MissingAttributeDetail> transmite información sobre atributos que son necesarios para la evaluación de la política y que faltaban en el contexto de petición.

```
<xs:element name="MissingAttributeDetail" type="xacml-
context:MissingAttributeDetailType"/>
<xs:complexType name="MissingAttributeDetailType">
<xs:sequence>
<xs:element ref="xacml-context:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
<xs:attribute name="DataType" type="xs:anyURI" use="required"/>
   <xs:attribute name="Issuer" type="xs:string" use="optional"/>
</xs:complexType>
```

El elemento <MissingAttributeDetail> es de tipo complejo **MissingAttributeDetailType**.

El elemento <MissingAttributeDetal> contiene los siguientes atributos y elementos:

– AttributeValue [Facultativo]

  El valor requerido del atributo que falta.

– <AttributeId> [Obligatorio]

  El identificador del atributo que falta.

– <DataType> [Obligatorio]

  El tipo de datos del atributo que falta.

– `Issuer` [Facultativo]

Este atributo, si se suministra, determinará el expedidor requerido del atributo que falta.

Si el PDP incluye elementos `<xacml-context:AttributeValue>` en el elemento `<MissingAttributeDetail>`, se trata de los valores aceptables para ese atributo. Si no se incluyen elementos `<xacml-context:AttributeValue>`, se trata de los nombres de atributos que el PDP no consiguió determinar durante su evaluación. La lista de atributos puede ser parcial o completa. El PDP no ofrece ninguna garantía de que sea suficiente con suministrar los valores o atributos que faltan para satisfacer la política.

7.6 Requisitos funcionales XACML

Esta cláusula especifica ciertos requisitos funcionales que no tienen que ver directamente con la producción o el consumo de un elemento XACML particular.

7.6.1 Punto de ejecución de la política (PEP)

Esta cláusula describe los requisitos para el PEP.

Una aplicación ejerce la función de PEP si mantiene el acceso a un conjunto de recursos y pide al PDP una decisión de autorización. El PEP debe observar las decisiones de autorización del PDP.

7.6.1.1 PEP de base

Si la decisión es "Permit", el PEP permitirá el acceso. Si la decisión está acompañada de obligaciones, el PEP sólo permitirá el acceso si comprende cuáles son esas obligaciones, puede cumplirlas y lo hace realmente.

Si la decisión es "Deny", el PEP denegará el acceso. Si la decisión está acompañada de obligaciones, el PEP sólo denegará el acceso si comprende cuáles son esas obligaciones, puede cumplirlas y lo hace realmente.

Si la decisión es "NotApplicable", no está definido el comportamiento del PEP.

Si la decisión es "Indeterminate", no está definido el comportamiento del PEP.

7.6.1.2 PEP con prioridad a la denegación

Si la decisión es "Permit", el PEP permitirá el acceso. Si la decisión está acompañada de obligaciones, el PEP sólo permitirá el acceso si comprende cuáles son esas obligaciones, puede cumplirlas y lo hace realmente.

Todas las demás decisiones se traducirán en una denegación del acceso.

NOTA – No se prohíben otras acciones, como la consulta de otros PDP, la repetición de formulaciones/sometimientos de la petición de decisión, etc.

7.6.1.3 PEP con prioridad al permiso

Si la decisión es "Deny", el PEP denegará el acceso. Si la decisión viene acompañada de obligaciones, el PEP sólo denegará el acceso si comprende cuáles son esas obligaciones, puede cumplirlas y lo hace realmente.

Todas las demás decisiones se traducirán en un permiso del acceso.

NOTA – No se prohíben otras acciones, como la consulta de otros PDP, la repetición de formulaciones/sometimientos de la petición de decisión, etc.

7.6.2 Evaluación de atributos

La función de servicio del contexto representa atributos en el contexto de petición, aunque no se hubieran incluido en la petición de decisión original, y se remite a ellos en la política mediante designadores y selectores de atributo de sujeto, recurso, acción y entorno. Los atributos designados son los criterios que utilizan determinados designadores y selectores de atributo de sujeto, recurso, acción y entorno para hacer referencia a determinados atributos en los respectivos elementos de sujeto, recurso, acción y entorno del contexto de petición.

7.6.2.1 Atributos estructurados

Los elementos `<xacml:AttributeValue>` y `<xacml-context:AttributeValue>` pueden contener un ejemplar de tipo de dato XML estructurado, como `<ds:KeyInfo>`. La presente Recomendación soporta varias formas de comparar el contenido de esos elementos.

1) En algunos casos, esos elementos se pueden comparar utilizando una de las funciones de cadena XACML, como "string-regexp-match", que se describen más adelante. Para ello es necesario especificar el siguiente tipo de datos para el elemento: "http://www.w3.org/2001/XMLSchema#string". Por ejemplo,

un tipo de dato estructurado que en realidad es un **ds:KeyInfo/KeyName** aparecería en el contexto como:

```
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
   &lt;ds:KeyName&gt;jhibbert-key&lt;/ds:KeyName&gt;
</AttributeValue>
```

En general, este método no resultará adecuado a menos que el tipo de datos estructurado sea muy simple.

2) Se puede utilizar un elemento `<AttributeSelector>` para seleccionar el contenido de un subelemento distal del tipo de dato estructurado mediante una expresión XPath. Posteriormente, ese valor se podrá comparar utilizando una de las funciones XACML soportadas que resulte adecuada para su tipo de dato primitivo. Para utilizar este método es necesario que el PDP soporte la función de expresiones XPath facultativa.

3) Se puede utilizar un elemento `<AttributeSelector>` para seleccionar cualquier nodo en el tipo de dato estructurado mediante una expresión XPath. Posteriormente, ese nodo se podrá comparar utilizando una de las funciones basadas en XPath que se describen en A.3. Para utilizar este método es necesario que el PDP soporte las expresiones XPath y las funciones XPath facultativas.

7.6.2.2 Multiconjuntos de atributos

XACML define conjuntos implícitos de sus tipos de datos. En XACML, un multiconjunto es un conjunto de valores que son del mismo tipo de datos. Los multiconjuntos de tipos de datos son necesarios debido a que las selecciones de nodos realizados a partir de un recurso XML o un contexto de petición XACML pueden devolver más de un valor.

El elemento `<AttributeSelector>` utiliza una expresión XPath para determinar la selección de datos a partir de un recurso XML. El resultado de una expresión XPath es un conjunto de nodos, que contiene todos los nodos distales determinados a partir del recurso XML que se ajustan al predicado de la expresión XPath. Teniendo en cuenta las distintas funciones de indexación que proporciona W3C XPath:1999, se supone que el conjunto de nodos resultante es el conjunto de los nodos concordantes. La presente Recomendación también define el elemento `<AttributeDesignator>` de forma que tenga la misma metodología de correspondencia para los atributos en el contexto de petición XACML.

Los valores de un multiconjunto no están ordenados, y algunos de ellos pueden estar duplicados. No podrá haber multiconjuntos que contengan otros multiconjuntos o valores de distintos tipos (es decir, un multiconjunto en XACML sólo puede contener valores que sean del mismo tipo de datos).

7.6.2.3 Atributos multivalor

Si un elemento `<Attribute>` en un contexto de petición contiene múltiples elementos derivados `<xacml-context:AttributeValue>`, el multiconjunto de valores resultante de la evaluación del elemento `<Attribute>` debe ser idéntico al multiconjunto de valores que se obtiene al evaluar un contexto en el que cada elemento `<xacml-context:AttributeValue>` aparece en un elemento `<Attribute>` separado, cada uno de los cuales transporta metadatos idénticos.

7.6.2.4 Correspondencia de atributos

Un atributo designado incluye criterios específicos para la concordancia de los atributos en el contexto. Un atributo determina el `AttributeId` y el `DataType`, y un atributo designado también especifica el `Issuer`. El atributo designado concordará con un atributo si los valores de sus respectivos `AttributeId`, `DataType` y los atributos opcionales de `Issuer` concuerdan dentro del elemento particular del contexto – sujeto, recurso, acción o entorno –. El `AttributeId` del atributo designado ha de corresponder (el mismo URI) con el `AttributeId` del correspondiente atributo del contexto. El `DataType` del atributo designado ha de corresponder (el mismo URI) con el `DataType` del correspondiente atributo del contexto. Si se indica el expedidor en el atributo designado, éste ha de corresponder, utilizando la función `urn:oasis:names:tc:xacml:1.0:function:string-equal`, con el `Issuer` del correspondiente atributo del contexto. Si no se indica el `Issuer` en el atributo designado, sólo el `AttributeId` y el `DataType` determinarán la correspondencia del atributo del contexto con el atributo designado, con independencia de si el `Issuer` está presente o ausente, o de cuál sea el valor real del expedidor en el correspondiente atributo del contexto. En el caso de un selector de atributos, la expresión XPath y el `DataType` determinarán la correspondencia del atributo con el atributo designado.

7.6.2.5 Recuperación de atributos

El PDP solicitará al controlador o función de servicio del contexto los valores de atributos del contexto de petición. El PDP indicará los atributos como si estuvieran en un documento físico del contexto de petición, pero el controlador del contexto debe obtener y proporcionar los valores solicitados por cualquier medio que estime apropiado. El controlador del contexto devolverá los valores de atributos que correspondan con el designador de atributos o el selector de

atributos y los reunirá en un multiconjunto de valores con el tipo de datos especificado. Si no hay concordancia con ningún atributo del contexto de petición, el atributo se considerará ausente. Si el atributo está ausente, el parámetro `MustBePresent` determinará si el designador de atributos o el selector de atributos devuelven un multiconjunto vacío o un resultado "Indeterminate". Si `MustBePresent` es "False" (valor por defecto), un atributo ausente se traducirá en un multiconjunto vacío. Si `MustBePresent` es "True", un atributo será "Indeterminate". Las consecuencias de este resultado "Indeterminate" dependerán de la especificación de las expresiones, reglas, políticas y conjuntos de políticas globales. Si el resultado es "Indeterminate", el `AttributeId`, el `DataType` y el `Issuer` del atributo podrán figurar en la decisión de autorización. No obstante, un PDP puede optar por no devolver esa información por motivos de seguridad.

7.6.2.6 Atributos del entorno

Si en la petición de decisión se suministra un valor para uno de los atributos del entorno, el controlador del contexto utilizará dicho valor. De no ser así, el controlador del contexto proporcionará un valor. En el caso de los atributos de fecha y hora, la semántica del valor suministrado indicará la "fecha y la hora de la petición de decisión".

7.6.3 Evaluación de la expresión

XACML determina expresiones en función de los elementos que se enumeran más adelante, de los cuales los elementos `<Apply>` y `<Condition>` componen expresiones más extensas de forma recursiva. Las expresiones serán válidas si son del tipo correcto, es decir, si los tipos de cada uno de los elementos contenidos en los elementos `<Apply>` y `<Condition>` corresponden a los respectivos tipos de argumento de la función designada por el atributo `FunctionId`. El tipo resultante de los elementos `<Apply>` o `<Condition>` será el tipo resultante de la función, que se puede reducir, mediante unificación del tipo, a un tipo de datos primitivo o a un multiconjunto de un tipo de datos primitivo. Uno de los resultados de evaluación de XACML "Indeterminate", que se considera como el resultado de una expresión inválida, o un error operacional ocurrido durante la evaluación de la expresión.

XACML define estos elementos en el grupo de sustitución del elemento `<Expression>`:

- `<xacml:AttributeValue>`
- `<xacml:SubjectAttributeDesignator>`
- `<xacml:ResourceAttributeDesignator>`
- `<xacml:ActionAttributeDesignator>`
- `<xacml:EnvironmentAttributeDesignator>`
- `<xacml:AttributeSelector>`
- `<xacml:Apply>`
- `<xacml:Condition>`
- `<xacml:Function>`
- `<xacml:VariableReference>`

7.6.4 Evaluación aritmética

En IEEE 754 se explica cómo evaluar funciones aritméticas en un contexto y especifican las opciones por defecto para precisión, redondeo, etc. XACML utilizará esta especificación para la evaluación de todas las funciones de entero y dobles, aplicando el contexto por defecto ampliado y el principio de doble precisión:

- banderas: todas puestas a 0,
- habilitadores de mensajes espontáneos (*trap*): todos puestos a 0 con excepción del habilitador "division-by-zero" (división por cero), que se pone a 1,
- precisión: se pone a la precisión doble designada,
- redondeo: opción de redondeo de valores mitad al entero par más próximo.

7.6.5 Evaluación de correspondencia

Los elementos de correspondencia de los atributos aparecen en el elemento `<Target>` de las reglas, las políticas y los conjuntos de políticas, y son los siguientes:

- `<SubjectMatch>`
- `<ResourceMatch>`
- `<ActionMatch>`
- `<EnvironmentMatch>`

Estos elementos representan expresiones booleanas sobre atributos del sujeto, el recurso, la acción y el entorno, respectivamente. Un elemento de correspondencia contiene un atributo `MatchId` que especifica la función que se va a utilizar para evaluar la concordancia, un `<xacml:AttributeValue>` y un elemento `<AttributeDesignator>` o `<AttributeSelector>` que definen el atributo del contexto que se va a comparar con el valor especificado.

El atributo `MatchId` determinará una función que realice la comparación entre dos argumentos, y devolverá un resultado del tipo "http://www.w3.org/2001/XMLSchema#boolean". El primer argumento del elemento de correspondencia especificará el valor del atributo para la función `MatchId`, y el segundo argumento será un elemento del multiconjunto devuelto por el elemento `<AttributeDesignator>` o `<AttributeSelector>`, tal como se explica más adelante. El `DataType` del `<xacml:AttributeValue>` deberá coincidir con el tipo de datos del primer argumento previsto para la función `MatchId`. El `DataType` del elemento `<AttributeDesignator>` o `<AttributeSelector>` deberá coincidir con el tipo de datos del segundo argumento previsto para la función `MatchId`.

Las funciones estándar XACML que cumplen los requisitos para su uso como valor de atributo de `MatchId` son las siguientes:

```
urn:oasis:names:tc:xacml:2.0:function:-type-equal
urn:oasis:names:tc:xacml:2.0:function:-type-greater-than
urn:oasis:names:tc:xacml:2.0:function:-type-greater-than-or-equal
urn:oasis:names:tc:xacml:2.0:function:-type-less-than
urn:oasis:names:tc:xacml:2.0:function:-type-less-than-or-equal
urn:oasis:names:tc:xacml:2.0:function:-type-match
```

Además, las funciones que están estrictamente dentro de una extensión de XACML pueden aparecer como un valor para el atributo `MatchId`, y esas funciones pueden utilizar tipos de datos que a su vez son extensiones, siempre que la función de extensión devuelva un resultado booleano y sólo tenga como entradas dos tipos básicos. Es necesario que la función utilizada como el valor para el atributo `MatchId` se pueda indizar. El uso de funciones no indizables o complejas puede dificultar la evaluación de peticiones de decisión.

La semántica de la evaluación para un elemento de correspondencia es del siguiente modo. Si se produce un error operacional mientras se efectúa la evaluación del elemento `<AttributeDesignator>` o `<AttributeSelector>`, el resultado de la expresión completa será "Indeterminate". Si el resultado de la evaluación del elemento `<AttributeDesignator>` o `<AttributeSelector>` es un multiconjunto vacío, el resultado de la expresión será "False". En los demás casos, la función `MatchId` se aplicará entre el `<xacml:AttributeValue>` y cada elemento del multiconjunto devuelto por el elemento `<AttributeDesignator>` o `<AttributeSelector>`. Si el resultado evaluación de una o más de esas aplicaciones de la función es "True", el resultado de la expresión completa será "True". En los demás casos, si al menos una de las aplicaciones de la función produce un resultado "Indeterminate", el resultado será "Indeterminate". Por último, si el resultado de evaluación de todas las aplicaciones de la función es "False", el resultado de la expresión completa será "False".

Asimismo, es posible expresar la semántica de un elemento de correspondencia del objetivo en una condición. Por ejemplo, la expresión de correspondencia del objetivo que compara un "Subject-name" (nombre del sujeto) que empiece con el nombre de "John" puede expresarse del siguiente modo:

```
<SubjectMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
       John.*
    </AttributeValue>
    <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
</SubjectMatch>
```

La misma semántica de correspondencia puede expresarse también como un elemento `<Apply>` en una condición utilizando la función "`urn:oasis:names:tc:xacml:1.0:function:any-of`", tal como se indica a continuación:

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
    <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match"/>
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
        John.*
    </AttributeValue>
    <SubjectAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
         DataType="http://www.w3.org/2001/XMLSchema#string"/>
</Apply>
```

7.6.6 Evaluación del objetivo

El valor del objetivo será "Concordancia" ("*Match*") si todos los sujetos, recursos, acciones y entornos especificados en el objetivo concuerdan con los valores del contexto de petición. Si alguno de los sujetos, recursos, acciones o entornos especificados en el objetivo es "Indeterminate", el objetivo será "Indeterminate". En los demás casos el objetivo será "Discordancia" ("*No-match*"). El cuadro 7-1 resume las condiciones de correspondencia del objetivo.

Cuadro 7-1/X.1142 – Cuadro de correspondencia del objetivo

Valor de los sujetos	Valor de los recursos	Valor de las acciones	Valor de los entornos	Valor del objetivo
"Concordancia"	"Concordancia"	"Concordancia"	"Concordancia"	"Concordancia"
"Discordancia"	"Concordancia" o "Discordancia"	"Concordancia" o "Discordancia"	"Concordancia" o "Discordancia"	"Discordancia"
"Concordancia" o "Discordancia"	"Discordancia"	"Concordancia" o "Discordancia"	"Concordancia" o "Discordancia"	"Discordancia"
"Concordancia" o "Discordancia"	"Concordancia" o "Discordancia"	"Discordancia"	"Concordancia" o "Discordancia"	"Discordancia"
"Concordancia" o "Discordancia"	"Concordancia" o "Discordancia"	"Concordancia" o "Discordancia"	"Discordancia"	"Discordancia"
"Indeterminate"	Intrascendente	Intrascendente	Intrascendente	"Indeterminate"
Intrascendente	"Indeterminate"	Intrascendente	Intrascendente	"Indeterminate"
Intrascendente	Intrascendente	"Indeterminate"	Intrascendente	"Indeterminate"
Intrascendente	Intrascendente	Intrascendente	"Indeterminate"	"Indeterminate"

Los sujetos, recursos, acciones y entornos coincidirán con valores en el contexto de petición si al menos uno de sus elementos `<Subject>`, `<Resource>`, `<Action>` o `<Environment>`, respectivamente, coincide con un valor en el contexto de petición. El cuadro 7-2 resume las condiciones de correspondencia de los sujetos. Los cuadros de correspondencia de los recursos, acciones y entornos son similares.

Cuadro 7-2/X.1142 – Cuadro de correspondencia de los sujetos

Valores `<Subject>`	**Valor** `<Subjects>`
Al menos uno es "Concordancia"	"Concordancia"
Ninguno concuerda y al menos uno es "Indeterminate"	"Indeterminate"
Todos "Discordancia"	"Discordancia"

Un sujeto, recurso, acción o entorno corresponderá con un valor en el contexto de petición si el valor de todos sus elementos `<SubjectMatch>`, `<ResourceMatch>`, `<ActionMatch>` o `<EnvironmentMatch>` respectivamente, es "True".

El cuadro 7-3 resume las condiciones de correspondencia del sujeto. Los cuadros de correspondencia del recurso, la acción y el entorno son similares.

Cuadro 7-3/X.1142 – Cuadro de correspondencia del sujeto

Valores `<SubjectMatch>`	**Valor** `<Subject>`
Todos "True"	"Concordancia"
Ninguno "False" y al menos uno "Indeterminate"	"Indeterminate"
Al menos uno "False"	"Discordancia"

7.6.7 Evaluación de VariableReference

El elemento `<VariableReference>` hace referencia a un único elemento `<VariableDefinition>` contenido en el mismo elemento `<Policy>`. Un elemento `<VariableReference>` que no indica ningún elemento `<VariableDefinition>` particular dentro del elemento global `<Policy>` es una referencia indefinida. Las políticas con referencias indefinidas no son válidas.

Cuando aparezca un <VariableReference>, será como si el texto del elemento <Expression> definido en el elemento <VariableDefinition> reemplazase al elemento <VariableReference>. Cualquier esquema de evaluación que conserve esta semántica es aceptable. Por ejemplo, es posible hacer una evaluación de la expresión en el elemento <VariableDefinition> y registrar el valor resultante en memoria intermedia (cached) para múltiples referencias sin ninguna consecuencia (es decir, el valor de un elemento <Expression> permanece invariable durante toda la evaluación de la política). Esta característica es una de las ventajas de XACML como lenguaje declarativo.

7.6.8 Evaluación de la condición

El valor de la condición será "True" si no se ha incluido el elemento <Condition>, o si el resultado de su evaluación es "True". Su valor será "False" si el resultado de evaluación del elemento <Condition> es "False". El valor de la condición será "Indeterminate" si el resultado de la evaluación de la expresión contenida en el elemento <Condition> es "Indeterminate".

7.6.9 Evaluación de la regla

Una regla tiene un valor que se puede calcular mediante la evaluación de su contenido. La evaluación de la regla implica realizar evaluaciones separadas del objetivo y la condición de la regla. El cuadro 7-4 resume las condiciones de verificación de la regla.

Cuadro 7-4/X.1142 – Cuadro de verificación de la regla

Objetivo	Condición	Valor de la regla
"Concordancia"	"True"	Efecto
"Concordancia"	"False"	"NotApplicable"
"Concordancia"	"Indeterminate"	"Indeterminate"
"Discordancia"	Intrascendente	"NotApplicable"
"Indeterminate"	Intrascendente	"Indeterminate"

Si el valor del objetivo es "Discordancia" o "Indeterminate", el valor de la regla será, respectivamente, "NotApplicable" o "Indeterminate", sea cual sea el valor de la condición. Por lo tanto, en estos casos no es necesario evaluar la condición.

Si el valor del objetivo es "Concordancia" y el valor de la condición es "True", el efecto especificado en el elemento global <Rule> determinará el valor de la regla.

7.6.10 Evaluación de la política

El valor de una política tan sólo se determina mediante su contenido, que se compara con el contenido del contexto de petición. El valor de una política se determinará mediante la evaluación del objetivo y las reglas de la política.

Para determinar la aplicabilidad de una política se evalúa su objetivo. Si en la evaluación del objetivo se obtiene "Concordancia", el valor de la política se determinará mediante la evaluación de las reglas de la política, de conformidad con el algoritmo de combinación de reglas especificado. Si en la evaluación del objetivo se obtiene "Discordancia", el valor de la política será "NotApplicable"; y si se obtiene "Indeterminate", será "Indeterminate".

El cuadro 7-5 resume las condiciones de verificación de la política.

Cuadro 7-5/X.1142 – Cuadro de verdad de la política

Objetivo	Valores de las reglas	Valor de la política
"Concordancia"	Al menos un valor de la regla es su Efecto	Especificado mediante el algoritmo de combinación de reglas
"Concordancia"	Todos los valores de la regla son "NotApplicable"	"NotApplicable"
"Concordancia"	Al menos un valor de la regla es "Indeterminate"	Especificado mediante el algoritmo de combinación de reglas
"Discordancia"	Intrascendente	"NotApplicable"
"Indeterminate"	Intrascendente	"Indeterminate"

El valor "Al menos un valor de la regla es su Efecto" significa que o bien no se ha incluido el elemento <Rule>, o bien hay una de las reglas de la política (o varias) que se aplica a la petición de decisión (es decir, que origina en respuesta el valor de su "Efecto"). El valor "Todos los valores de la regla son "NotApplicable"' se utiliza si ninguna de las reglas de

la política se aplica a la petición, y ninguna origina en respuesta un valor de "Indeterminate". Si no hay ninguna regla de la política que se aplique a la petición, pero una o más reglas devuelven un valor de "Indeterminate", el resultado de la evaluación de las reglas será "Al menos un valor de la regla es 'Indeterminate'".

Si el valor del objetivo es "Discordancia" o "Indeterminate", el valor de la política será respectivamente "NotApplicable" o "Indeterminate", sea cual sea el valor de las reglas. Por lo tanto, en estos casos no es necesario evaluar las reglas.

Si el valor del objetivo es "Concordancia" y el valor de la regla es "Al menos un valor de la regla es su Efecto" o "Al menos un valor de la regla es 'Indeterminate'", el algoritmo de combinación de reglas especificado en la política determinará el valor de la política.

Obsérvese que ninguno de los algoritmos de combinación de reglas definidos en la presente Recomendación tiene parámetros. Sin embargo, los algoritmos de combinación que están normalizados sí pueden tomar parámetros. En tales casos, los valores de esos parámetros asociados con las reglas deben tenerse en cuenta cuando se evalúe la política. Es necesario, pues, que los parámetros y sus tipos estén definidos en la especificación del algoritmo de combinación. Si la implementación soporta los parámetros del combinador y se crea una política que incluye este tipo de parámetros, deberán suministrarse los valores de los parámetros para la implementación del algoritmo de combinación.

7.6.11 Evaluación de un conjunto de políticas

El valor de un conjunto de políticas se determina mediante su contenido, que se compara con el contenido del contexto de petición. El valor de un conjunto de políticas se determinará mediante la evaluación del objetivo del conjunto de políticas, las políticas y los conjuntos de políticas, de conformidad con el algoritmo de combinación de políticas especificado.

Para determinar la aplicabilidad de un conjunto de políticas se evalúa su objetivo. Si al evaluar el objetivo se obtiene "Concordancia", el valor del conjunto de políticas se determinará mediante la evaluación de las políticas y conjuntos de políticas que constituyen el conjunto de políticas, de conformidad con el algoritmo de combinación de políticas especificado. Si en la evaluación del objetivo se obtiene "Discordancia", el valor del conjunto de políticas será "NotApplicable"; y si se obtiene "Indeterminate", será "Indeterminate".

El cuadro 7-6 resume las condiciones de verificación del conjunto de políticas.

Cuadro 7.6/X.1142 – Cuadro de verificación del conjunto de políticas

Objetivo	Valores de las políticas	Valor del conjunto de políticas
"Concordancia"	Al menos un valor de las políticas es su Decisión	Especificado mediante el algoritmo de combinación de políticas
"Concordancia"	Todos los valores de las políticas son "NotApplicable"	"NotApplicable"
"Concordancia"	Al menos un valor de las políticas es "Indeterminate"	Especificado mediante el algoritmo de combinación de políticas
"Discordancia"	Intrascendente	"NotApplicable"
"Indeterminate"	Intrascendente	"Indeterminate"

El valor "Al menos un valor de las políticas es su Decisión" se utilizará si no hay ninguna política ni conjunto de políticas contenidas o referenciadas, y cuando haya una política o conjunto de políticas (o varias) dentro del conjunto de políticas o referenciadas por éste que sea aplicable a la petición de decisión (es decir, que origina en respuesta un valor determinado por su algoritmo de combinación). Se utilizará el valor "Todos los valores de las políticas son 'NotApplicable'" si no hay ninguna política ni conjunto de políticas dentro del conjunto de políticas o referenciados por éste que sean aplicables a la petición, y cuando no haya ninguna política ni conjunto de políticas dentro del conjunto de políticas o referenciados por éste que originen en respuesta un valor de "Indeterminate". Si no hay ninguna política ni conjunto de políticas dentro del conjunto de políticas o referenciados por éste que sean aplicables a la petición, pero una o más políticas o conjunto de políticas originen en respuesta un valor de "Indeterminate", el resultado de evaluación de las políticas será "Al menos un valor de las políticas es 'Indeterminate'".

Si el valor del objetivo es "Discordancia" o "Indeterminate", el valor del conjunto de políticas será respectivamente "NotApplicable" o "Indeterminate", sea cual sea el valor de las políticas. Por lo tanto, en estos casos no es necesario evaluar las políticas.

Si el valor del objetivo es "Concordancia" y el valor de las políticas es "Al menos un valor de las políticas es su Decisión" o "Al menos un valor de las políticas es 'Indeterminate'", el algoritmo de combinación de políticas especificado en el conjunto de políticas determinará el valor del conjunto de políticas.

Obsérvese que ninguno de los algoritmos de combinación de políticas definidos por XACML 2.0 tiene parámetros. Sin embargo, los algoritmos de combinación que no están normalizados sí pueden tener parámetros. En tales casos, los valores de esos parámetros asociados con las políticas deben tenerse en cuenta cuando se evalúe el conjunto de políticas. Es necesario, pues, que los parámetros y sus tipos estén definidos en la especificación del algoritmo de combinación. Si la implementación soporta los parámetros del combinador y se crea una política que incluye este tipo de parámetros, deberán suministrarse los valores de los parámetros para la implementación del algoritmo de combinación.

7.6.12 Recursos jerárquicos

Con frecuencia un recurso se organiza como una jerarquía (por ejemplo sistema de ficheros, documento XML). XACML proporciona varios mecanismos facultativos para soportar los recursos jerárquicos que se presentan en esta Recomendación.

7.6.13 Decisión sobre autorización

Con respecto a una solicitud de decisión, el PDP consiste en un algoritmo que combina varias políticas y en una serie de políticas y/o conjuntos de políticas. El PDP devolverá un contexto de respuesta como si hubiese evaluado un único conjunto de políticas consistente en dicho algoritmo de combinación y el conjunto de políticas y/o conjuntos de políticas.

El PDP debe evaluar el conjunto de políticas como se especifica en 7.4 y en 7.6. El PDP debe devolver un contexto de respuesta con un elemento de `<Decision>` cuyo valor será "Permitir", "Denegar", "Intedeterminate" o "NotApplicable".

Si el PDP no puede tomar una decisión, entonces se devolverá un elemento `<Decision>` "Indeterminate".

7.6.14 Obligaciones

Una política o conjunto de políticas puede contener una o más obligaciones. Cuando se evalúa dicha política o conjunto de políticas, la condición para elevar una obligación al siguiente nivel de la evaluación (la política que incluye o que remite a ésta, el conjunto de políticas o la decisión de autorización) es que el efecto de la política o conjunto de políticas que están siendo evaluados corresponda al valor del atributo `FulfillOn` de la obligación.

Como resultado de dicho procedimiento, no se devolverá ninguna obligación al PEP si las políticas o conjuntos de políticas en las que se basan no son evaluadas, o bien si el resultado de evaluación es "Indeterminate" o "NotApplicable", o si la decisión derivada de la evaluación de la política o conjunto de políticas no corresponde a la decisión resultante de la evaluación de un conjunto de políticas que incluye los otros.

Si se considera que la evaluación del PDP es un árbol de conjuntos de políticas y políticas, cada una de las cuales devuelve "Permitir" o "Denegar", entonces el conjunto de obligaciones devueltas por el PDP al PEP sólo incluirá las obligaciones asociadas con trayectos en los que se obtiene, en todos los niveles de evaluación, el mismo efecto que devuelve el PDP. Cuando no se pueda admitir ninguna falta de determinismo, se deberá utilizar un algoritmo de combinación determinístico, por ejemplo el algoritmo denegación prioritaria por orden.

7.6.15 Tratamiento de las excepciones

XACML especifica el comportamiento para el PDP en las siguientes situaciones.

7.6.15.1 Funcionalidad no soportada

Si el PDP intenta evaluar un conjunto de políticas o una política que contienen un tipo de elemento o una función opcionales que el PDP no soporta, entonces el PDP devolverá el valor de `<Decision>` "Indeterminate". Si se devuelve también un elemento `<StatusCode>`, entonces su valor será "`urn:oasis:names:tc:xacml:1.0:status:syntax-error`" (elemento no soportado) o "`urn:oasis:names:tc:xacml:1.0:status:processing-error`" (función no soportada).

7.6.15.2 Errores de sintaxis y de tipo

Si el PDP de XACML evalúa una política que contiene una sintaxis que no es válida, cuando recibe una petición de decisión, el resultado de esa política será "Indeterminate" con valor de `StatusCode`:

```
"urn:oasis:names:tc:xacml:1.0:status:syntax-error"
```

Si el PDP de XACML evalúa una política que contiene tipos de datos estáticos que no son válidos, cuando recibe una petición de decisión, el resultado de esa política será "Indeterminate" con valor de `StatusCode`:

```
"urn:oasis:names:tc:xacml:1.0:status:processing-error"
```

7.6.15.3 Atributos que faltan

Si el contexto de petición no contiene atributos concordantes con alguno de los designadores o seleccionadores de atributos de la política, el valor del elemento `<Decision>` será "Indeterminate". Si en este caso se proporciona además un código de estado, entonces se utilizará el valor:

```
"urn:oasis:names:tc:xacml:1.0:status:missing-attribute",
```

a fin de indicar que se necesita información adicional para alcanzar una decisión definitiva. En este caso, el elemento `<Status>` puede enumerar los nombres y los tipos de datos de cualquier atributo de los sujetos, el recurso, la acción o el entorno que el PDP precisa para poder tomar una decisión. Un PEP puede volver a presentar un contexto de petición más completo cuando recibe un elemento `<Decision>` "Indeterminate" con un código de estado:

```
"urn:oasis:names:tc:xacml:1.0:missing-attribute",
```

añadiendo valores de atributo para los nombres de atributo enumerados en la respuesta anterior. Cuando del PDP devuelve un elemento `<Decision>` "Indeterminate" con un código de estado:

```
"urn:oasis:names:tc:xacml:1.0:missing-attribute",
```

no debe enumerar los nombres ni los tipos de datos de ningún atributo del sujeto, el recurso, la acción o el entorno para los que se proporcionaron valores en la solicitud original. Obsérvese que esta condición obliga al PDP a devolver finalmente una decisión de autorización "Permitir", "Denegar" o "Indeterminate" con otro código de estado, como respuesta a cada una de las sucesivas peticiones más detalladas.

7.7 Puntos de extensibilidad XACML

Esta cláusula describe, a título informativo, los puntos del modelo y esquema XACML en los que se pueden añadir extensiones.

7.7.1 Tipos de atributo XML extensibles

Los siguientes atributos XML poseen valores que son URI. Estos pueden ser extendidos mediante la creación de nuevos URI asociados con nueva semántica para dichos atributos.

- `AttributeId`,
- `DataType`,
- `FunctionId`,
- `MatchId`,
- `ObligationId`,
- `PolicyCombiningAlgId`,
- `RuleCombiningAlgId`,
- `StatusCode`,
- `SubjectCategory`.

7.7.2 Atributos estructurados

Los elementos `<xacml:AttributeValue>` y `<xacml-context:AttributeValue>` pueden contener un ejemplar de un tipo de datos XML estructurado. A continuación se enumeran algunas técnicas que requieren extensiones XACML:

1) Para un determinado tipo de datos estructurado, una comunidad de usuarios XACML puede definir nuevos identificadores de atributos para cada subelemento distal del tipo de datos estructurado que sea conforme a uno de los tipos de datos primitivos definidos en XACML. Mediante estos nuevos identificadores de atributos, los PEP o las funciones de servicio de contexto utilizados por la citada comunidad de usuarios pueden asimilar los ejemplares del tipo de datos estructurado a una secuencia de elementos `<Attribute>` individuales. Cada uno de estos elementos `<Attribute>` puede compararse mediante las funciones definidas según XACML. Cuando se utiliza este método, el tipo de datos estructurado nunca aparecerá como tal en un elemento `<xacml-context:AttributeValue>`.
2) Una comunidad de usuarios XACML puede definir una nueva función que puede usarse para comparar un valor del tipo de datos estructurado con cualquier otro valor. Este método sólo puede ser utilizado por los PDP que soportan la nueva función.

7.8 Conformidad

El lenguaje XACML define varias funciones de uso un tanto especial, que no son de aplicación obligatoria para que la implementación sea conforme a esta Recomendación.

La presente cláusula enumera aquellas partes de esta Recomendación que deben incluirse en la implementación de un PDP para que el sistema sea conforme a la norma XACML v2.0.

NOTA – "M" significa de implementación obligatoria. "O" significa facultativo.

7.8.1 Elementos del esquema

La implementación debe soportar aquellos elementos del esquema marcados con "M".

Nombre del elemento	M/O
xacml-context:Action	M
xacml-context:Attribute	M
xacml-context:AttributeValue	M
xacml-context:Decision	M
xacml-context:Environment	M
xacml-context:MissingAttributeDetail	M
xacml-context:Obligations	O
xacml-context:Request	M
xacml-context:Resource	M
xacml-context:ResourceContent	O
xacml-context:Response	M
xacml-context:Result	M
xacml-context:Status	M
xacml-context:StatusCode	M
xacml-context:StatusDetail	O
xacml-context:StatusMessage	O
xacml-context:Subject	M
xacml:Action	M
xacml:ActionAttributeDesignator	M
xacml:ActionMatch	M
xacml:Actions	M
xacml:Apply	M
xacml:AttributeAssignment	O
xacml:AttributeSelector	O
xacml:AttributeValue	M
xacml:CombinerParameters	O
xacml:CombinerParameter	O
xacml:Condition	M
xacml:Description	M
xacml:Environment	M
xacml:EnvironmentMatch	M
xacml:EnvironmentAttributeDesignator	M
xacml:Environments	M
xacml:Expression	M
xacml:Function	M
xacml:Obligation	O
xacml:Obligations	O
xacml:Policy	M
xacml:PolicyCombinerParameters	O
xacml:PolicyDefaults	O
xacml:PolicyIdReference	M
xacml:PolicySet	M

Nombre del elemento	M/O
xacml:PolicySetDefaults	O
xacml:PolicySetIdReference	M
xacml:Resource	M
xacml:ResourceAttributeDesignator	M
xacml:ResourceMatch	M
xacml:Resources	M
xacml:Rule	M
xacml:RuleCombinerParameters	O
xacml:Subject	M
xacml:SubjectMatch	M
xacml:Subjects	M
xacml:Target	M
xacml:VariableDefinition	M
xacml:VariableReference	M
xacml:XPathVersion	O

7.8.2 Prefijos de identificador

Los siguientes prefijos de identificador se reservan para el XACML.

Identificador
urn:oasis:names:tc:xacml:2.0
urn:oasis:names:tc:xacml:2.0:conformance-test
urn:oasis:names:tc:xacml:2.0:context
urn:oasis:names:tc:xacml:2.0:example
urn:oasis:names:tc:xacml:1.0:function
urn:oasis:names:tc:xacml:2.0:function
urn:oasis:names:tc:xacml:2.0:policy
urn:oasis:names:tc:xacml:1.0:subject
urn:oasis:names:tc:xacml:1.0:resource
urn:oasis:names:tc:xacml:1.0:action
urn:oasis:names:tc:xacml:1.0:environment
urn:oasis:names:tc:xacml:1.0:status

7.8.3 Algoritmos

La implementación debe incluir los algoritmos de combinación de reglas y políticas asociados a los siguientes identificadores marcados con la letra "M".

Algoritmo	M/O
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides	M
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides	M
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides	M
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides	M
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable	M
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable	M
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:only-one-applicable	M
urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-deny-overrides	M
urn:oasis:names:tc:xacml:1.1:policy-combining-algorithm:ordered-deny-overrides	M
urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-permit-overrides	M
urn:oasis:names:tc:xacml:1.1:policy-combining-algorithm:ordered-permit-overrides	M

7.8.4 Códigos de estado

Si la implementación soporta el elemento <StatusCode> (es facultativo), se deberán soportar los siguientes códigos de estado, utilizándolos según la manera especificada en XACML.

Identificador	M/O
urn:oasis:names:tc:xacml:1.0:status:missing-attribute	M
urn:oasis:names:tc:xacml:1.0:status:ok	M
urn:oasis:names:tc:xacml:1.0:status:processing-error	M
urn:oasis:names:tc:xacml:1.0:status:syntax-error	M

7.8.5 Atributos

La implementación debe soportar los atributos asociados a los siguientes identificadores especificados según XACML. Si los valores para dichos atributos no se encuentran en la petición de decisión, deberá proporcionarlos la función de servicio de contexto. Por consiguiente, a diferencia de la mayoría de otros atributos, su semántica no es transparente para el PDP.

Identificador	M/O
urn:oasis:names:tc:xacml:1.0:environment:current-time	M
urn:oasis:names:tc:xacml:1.0:environment:current-date	M
urn:oasis:names:tc:xacml:1.0:environment:current-dateTime	M

7.8.6 Identificadores

La implementación debe utilizar los atributos asociados a los siguientes identificadores según la manera especificada en XACML. Este requisito tiene que ver fundamentalmente con la implementación de un PAP o PEP que utiliza XACML, dado que la semántica de los atributos es transparente para el PDP.

Identificador	M/O
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:dns-name	O
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:ip-address	O
urn:oasis:names:tc:xacml:1.0:subject:authentication-method	O
urn:oasis:names:tc:xacml:1.0:subject:authentication-time	O
urn:oasis:names:tc:xacml:1.0:subject:key-info	O
urn:oasis:names:tc:xacml:1.0:subject:request-time	O
urn:oasis:names:tc:xacml:1.0:subject:session-start-time	O
urn:oasis:names:tc:xacml:1.0:subject:subject-id	O
urn:oasis:names:tc:xacml:1.0:subject:subject-id-qualifier	O
urn:oasis:names:tc:xacml:1.0:subject-category:access-subject	M
urn:oasis:names:tc:xacml:1.0:subject-category:codebase	O
urn:oasis:names:tc:xacml:1.0:subject-category:intermediary-subject	O
urn:oasis:names:tc:xacml:1.0:subject-category:recipient-subject	O
urn:oasis:names:tc:xacml:1.0:subject-category:requesting-machine	O
urn:oasis:names:tc:xacml:1.0:resource:resource-location	O
urn:oasis:names:tc:xacml:1.0:resource:resource-id	M
urn:oasis:names:tc:xacml:1.0:resource:simple-file-name	O
urn:oasis:names:tc:xacml:1.0:action:action-id	O
urn:oasis:names:tc:xacml:1.0:action:implied-action	O

7.8.7 Tipos de datos

La implementación deber soportar los tipos de datos asociados a los siguientes identificadores marcados con "M".

Tipos de datos	M/O
http://www.w3.org/2001/XMLSchema#string	M
http://www.w3.org/2001/XMLSchema#boolean	M
http://www.w3.org/2001/XMLSchema#integer	M
http://www.w3.org/2001/XMLSchema#double	M
http://www.w3.org/2001/XMLSchema#time	M
http://www.w3.org/2001/XMLSchema#date	M
http://www.w3.org/2001/XMLSchema#dateTime	M
urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration	M
urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration	M
http://www.w3.org/2001/XMLSchema#anyURI	M
http://www.w3.org/2001/XMLSchema#hexBinary	M
http://www.w3.org/2001/XMLSchema#base64Binary	M
urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name	M
urn:oasis:names:tc:xacml:1.0:data-type:x500Name	M

7.8.8 Funciones

La implementación debe procesar adecuadamente aquellas funciones asociadas a los identificadores marcados con "M".

Función	M/O
urn:oasis:names:tc:xacml:1.0:function:string-equal	M
urn:oasis:names:tc:xacml:1.0:function:boolean-equal	M
urn:oasis:names:tc:xacml:1.0:function:integer-equal	M
urn:oasis:names:tc:xacml:1.0:function:double-equal	M
urn:oasis:names:tc:xacml:1.0:function:date-equal	M
urn:oasis:names:tc:xacml:1.0:function:time-equal	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-equal	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-equal	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-equal	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-equal	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-equal	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-equal	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-equal	M
urn:oasis:names:tc:xacml:1.0:function:integer-add	M
urn:oasis:names:tc:xacml:1.0:function:double-add	M
urn:oasis:names:tc:xacml:1.0:function:integer-subtract	M
urn:oasis:names:tc:xacml:1.0:function:double-subtract	M
urn:oasis:names:tc:xacml:1.0:function:integer-multiply	M
urn:oasis:names:tc:xacml:1.0:function:double-multiply	M
urn:oasis:names:tc:xacml:1.0:function:integer-divide	M
urn:oasis:names:tc:xacml:1.0:function:double-divide	M
urn:oasis:names:tc:xacml:1.0:function:integer-mod	M
urn:oasis:names:tc:xacml:1.0:function:integer-abs	M
urn:oasis:names:tc:xacml:1.0:function:double-abs	M
urn:oasis:names:tc:xacml:1.0:function:round	M
urn:oasis:names:tc:xacml:1.0:function:floor	M
urn:oasis:names:tc:xacml:1.0:function:string-normalize-space	M
urn:oasis:names:tc:xacml:1.0:function:string-normalize-to-lower-case	M

Función	M/O
urn:oasis:names:tc:xacml:1.0:function:double-to-integer	M
urn:oasis:names:tc:xacml:1.0:function:integer-to-double	M
urn:oasis:names:tc:xacml:1.0:function:or	M
urn:oasis:names:tc:xacml:1.0:function:and	M
urn:oasis:names:tc:xacml:1.0:function:n-of	M
urn:oasis:names:tc:xacml:1.0:function:not	M
urn:oasis:names:tc:xacml:1.0:function:integer-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:integer-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:integer-less-than	M
urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:double-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:double-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:double-less-than	M
urn:oasis:names:tc:xacml:1.0:function:double-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-add-dayTimeDuration	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-add-yearMonthDuration	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-subtract-dayTimeDuration	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-subtract-yearMonthDuration	M
urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration	M
urn:oasis:names:tc:xacml:1.0:function:date-subtract-yearMonthDuration	M
urn:oasis:names:tc:xacml:1.0:function:string-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:string-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:string-less-than	M
urn:oasis:names:tc:xacml:1.0:function:string-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:time-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:time-less-than	M
urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal	M
urn:oasis:names:tc:xacml:2.0:function:time-in-range	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:date-greater-than	M
urn:oasis:names:tc:xacml:1.0:function:date-greater-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:date-less-than	M
urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal	M
urn:oasis:names:tc:xacml:1.0:function:string-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:string-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:string-is-in	M
urn:oasis:names:tc:xacml:1.0:function:string-bag	M
urn:oasis:names:tc:xacml:1.0:function:boolean-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:boolean-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:boolean-is-in	M
urn:oasis:names:tc:xacml:1.0:function:boolean-bag	M
urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:integer-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:integer-is-in	M
urn:oasis:names:tc:xacml:1.0:function:integer-bag	M
urn:oasis:names:tc:xacml:1.0:function:double-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:double-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:double-is-in	M

Función	M/O
urn:oasis:names:tc:xacml:1.0:function:double-bag	M
urn:oasis:names:tc:xacml:1.0:function:time-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:time-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:time-is-in	M
urn:oasis:names:tc:xacml:1.0:function:time-bag	M
urn:oasis:names:tc:xacml:1.0:function:date-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:date-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:date-is-in	M
urn:oasis:names:tc:xacml:1.0:function:date-bag	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-is-in	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-bag	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-is-in	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-bag	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-is-in	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-bag	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-is-in	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-bag	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-is-in	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-bag	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-is-in	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-bag	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-is-in	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-bag	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-one-and-only	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-bag-size	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-is-in	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-bag	M
urn:oasis:names:tc:xacml:2.0:function:string-concatenate	M
urn:oasis:names:tc:xacml:2.0:function:uri-string-concatenate	M
urn:oasis:names:tc:xacml:1.0:function:any-of	M
urn:oasis:names:tc:xacml:1.0:function:all-of	M
urn:oasis:names:tc:xacml:1.0:function:any-of-any	M
urn:oasis:names:tc:xacml:1.0:function:all-of-any	M
urn:oasis:names:tc:xacml:1.0:function:any-of-all	M
urn:oasis:names:tc:xacml:1.0:function:all-of-all	M
urn:oasis:names:tc:xacml:1.0:function:map	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-match	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match	M

Función	M/O
urn:oasis:names:tc:xacml:1.0:function:string-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:anyURI-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:ipAddress-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:dnsName-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:rfc822Name-regexp-match	M
urn:oasis:names:tc:xacml:2.0:function:x500Name-regexp-match	M
urn:oasis:names:tc:xacml:1.0:function:xpath-node-count	O
urn:oasis:names:tc:xacml:1.0:function:xpath-node-equal	O
urn:oasis:names:tc:xacml:1.0:function:xpath-node-match	O
urn:oasis:names:tc:xacml:1.0:function:string-intersection	M
urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:string-union	M
urn:oasis:names:tc:xacml:1.0:function:string-subset	M
urn:oasis:names:tc:xacml:1.0:function:string-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:boolean-intersection	M
urn:oasis:names:tc:xacml:1.0:function:boolean-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:boolean-union	M
urn:oasis:names:tc:xacml:1.0:function:boolean-subset	M
urn:oasis:names:tc:xacml:1.0:function:boolean-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:integer-intersection	M
urn:oasis:names:tc:xacml:1.0:function:integer-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:integer-union	M
urn:oasis:names:tc:xacml:1.0:function:integer-subset	M
urn:oasis:names:tc:xacml:1.0:function:integer-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:double-intersection	M
urn:oasis:names:tc:xacml:1.0:function:double-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:double-union	M
urn:oasis:names:tc:xacml:1.0:function:double-subset	M
urn:oasis:names:tc:xacml:1.0:function:double-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:time-intersection	M
urn:oasis:names:tc:xacml:1.0:function:time-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:time-union	M
urn:oasis:names:tc:xacml:1.0:function:time-subset	M
urn:oasis:names:tc:xacml:1.0:function:time-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:date-intersection	M
urn:oasis:names:tc:xacml:1.0:function:date-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:date-union	M
urn:oasis:names:tc:xacml:1.0:function:date-subset	M
urn:oasis:names:tc:xacml:1.0:function:date-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-intersection	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-union	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-subset	M
urn:oasis:names:tc:xacml:1.0:function:dateTime-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-intersection	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-union	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-subset	M
urn:oasis:names:tc:xacml:1.0:function:anyURI-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-intersection	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-union	M

Función	M/O
urn:oasis:names:tc:xacml:1.0:function:hexBinary-subset	M
urn:oasis:names:tc:xacml:1.0:function:hexBinary-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-intersection	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-union	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-subset	M
urn:oasis:names:tc:xacml:1.0:function:base64Binary-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-intersection	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-union	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-subset	M
urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-intersection	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-union	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-subset	M
urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-intersection	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-union	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-subset	M
urn:oasis:names:tc:xacml:1.0:function:x500Name-set-equals	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-intersection	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-at-least-one-member-of	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-union	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-subset	M
urn:oasis:names:tc:xacml:1.0:function:rfc822Name-set-equals	M

8 Perfil de control de acceso basado en el cometido jerárquico fundamental (RBAC)

En esta cláusula se define un perfil para el uso del XACML que satisface los requisitos para un sistema de control de acceso basado en el cometido (RBAC), jerárquico y fundamental.

8.1 Introducción al RBAC

Esta cláusula es informativa.

En esta cláusula se define un perfil para el XACML, que satisface los requisitos para un sistema de control de acceso basado en el cometido (RBAC), jerárquico y fundamental.

NOTA – Véase [RBAC] para obtener información sobre el RBAC.

8.1.1 Alcance

El control de acceso basado en el cometido permite especificar políticas especificando los cometidos del sujeto y no únicamente la identidad de cada sujeto. Esto es importante a efectos de escalabilidad y gestión de sistemas de control de acceso.

Mediante las políticas especificadas en este perfil se puede responder a tres tipos de preguntas:

1) Si un sujeto tiene los cometidos R_1, R_2, ... R_n habilitados, ¿puede el sujeto X acceder a un recurso dado mediante una acción determinada?

2) ¿El sujeto X está autorizado a tener el cometido R_i habilitado?

3) Si un sujeto tiene los cometidos R_1 , R_2, ... R_n habilitados, ¿significa esto que el sujeto tendrá permisos asociados a un cometido dado R'? Es decir, ¿es el cometido R' igual a cualquiera de los cometidos R_1, R_2, ... R_n o *dependiente* de ellos?

Las políticas especificadas en este perfil no responden a la pregunta: ¿Qué cometidos tiene el sujeto X? Dicha pregunta debe canalizarse a través de una autoridad de habilitación de cometidos y no la trata directamente un PDP en XACML. Dicha entidad podrá hacer uso de las políticas de XACML, si bien necesitará información adicional.

En las políticas especificadas en este perfil se supone que todos los cometidos para un sujeto dado ya están habilitados cuando se solicita una decisión de autorización. No se aplican a entornos en que los cometidos tienen que habilitarse dinámicamente basándose en el recurso o acciones que está intentando llevar a cabo un sujeto. Por esta razón, las políticas especificadas en este perfil tampoco incluyen una "separación de funciones" estática o dinámica. Ulteriormente se podrá definir un perfil adaptado a los requisitos de este tipo de entorno.

8.1.2 Cometido

En esta Recomendación, los cometidos se expresan como atributos de sujeto XACML. Existen dos excepciones: en una `<PolicySet>` o `<Policy>` Asignación de Cometido y en una `<Policy> HasPrivilegesOfRole`, el cometido aparece como un atributo de recurso.

Los atributos de cometido pueden expresarse de dos formas, dependiendo de los requisitos del entorno de aplicación. En algunos entornos puede haber un pequeño número de "atributos de cometido", cuyo nombre es un término que denota "cometido", y cuyo valor indica el nombre del cometido mantenido. Por ejemplo, en este primer tipo de entorno, puede haber un "atributo de cometido" que tiene el `AttributeId "&role;"` (este perfil recomienda el uso de este identificador). Los cometidos posibles son valores para este atributo y pueden ser "`&roles;officer`", "`&roles;manager`", y "`&roles;employee`". Esta forma de expresar cometidos funciona mejor con el modo de indicar políticas del XACML. Este método de identificación de cometidos también contribuye más a la interoperabilidad.

Asimismo, en otros entornos de aplicación puede haber varios identificadores de atributo diferentes, indicando cada uno de ellos un cometido diferente. Por ejemplo, en este segundo tipo de entorno puede haber tres identificadores de atributos: "`urn:someapp:attributes:officer-role`", "`urn:someapp:attributes:manager-role`", y "`urn:someapp:attributes:employee-role`". En este caso el valor del atributo puede estar vacío o contener varios parámetros asociados al cometido. Las políticas del XACML pueden tratar cometidos expresados de esta manera, pero la primera solución es más natural.

XACML soporta múltiples sujetos por petición de acceso, indicando varias entidades que pueden participar en dicha petición. Por ejemplo, normalmente hay un usuario humano que inicia la petición, al menos indirectamente. Normalmente hay una o varias aplicaciones o bases de código que generan la petición de acceso de bajo nivel propiamente dicha en nombre del usuario. Hay algún dispositivo de computación en el que se ejecuta la aplicación o base de código, pudiendo tener dicho dispositivo una identidad tal como una dirección IP. XACML identifica a cada uno de dichos `Subject` mediante un atributo xml `SubjectCategory` que indica el tipo de sujeto descrito. Por ejemplo, la `SubjectCategory` del usuario humano es `&subject-category;access-subject` (categoría por defecto); la aplicación que genera la petición de acceso tiene una `SubjectCategory` de `&subject-category;codebase`, y así sucesivamente. En este perfil, un atributo de cometido puede asociarse a cualquiera de las categorías de sujetos que forman parte de la petición de acceso.

8.1.3 Políticas

En esta Recomendación se especifican cuatro tipos de políticas.

1) El `<PolicySet>` **Cometido** o **RPS**: Un `<PolicySet>` que asocia a los titulares de un atributo de cometido y un valor dados con un `<PolicySet>` Permiso que contiene los permisos propiamente dichos asociados con ese cometido. El elemento `<Target>` de un `<PolicySet>` Cometido limita la aplicabilidad del `<PolicySet>` a los sujetos poseedores del atributo y valor del cometido asociados. Cada `<PolicySet>` Cometido remite a un único `<PolicySet>` Permiso, pero no contiene ningún otro elemento `<Policy>` o `<PolicySet>` ni remite a estos elementos.

2) El `<PolicySet>` **Permiso** o **PPS**: Un `<PolicySet>` que contiene los permisos propiamente dichos asociados a un cometido dado. Contiene elementos `<Policy>` y `<Rules>` que describen los recursos y acciones a los que se permite acceder a los sujetos, así como cualquier otra condición acerca de dicho acceso, por ejemplo la hora. Un `<PolicySet>` Permiso también puede contener referencias a otros `<PolicySet>` Permiso asociados a otros cometidos de rango inferior al cometido dado, permitiendo así que el primer `<PolicySet>` Permiso herede todos los permisos asociados con el cometido del `<PolicySet>` Permiso referenciado. El elemento `<Target>` de un `<PolicySet>` Permiso, si está presente, no debe limitar los sujetos a los que es aplicable el `<PolicySet>`.

3) **Asignación de Cometidos** <Policy> o <PolicySet>: Una <Policy> o <PolicySet> que define qué cometidos pueden ser habilitados o asignados a qué sujetos. También puede especificar restricciones sobre combinaciones de cometidos o números totales de cometidos asignados o habilitados para un sujeto dado. Este tipo de política es el que utiliza la autoridad de habilitación de cometidos. El uso de una <Policy> o <PolicySet> Asignación de Cometidos es facultativo.

4) **HasPrivilegesOfRole** <Policy>: Una <Policy> de una <PolicySet> Permiso que soporta peticiones sobre si un sujeto tiene los privilegios asociados a un cometido dado. Si el sistema soporta este tipo de petición, es preciso incluir la <Policy> HasPrivilegesOfRole en cada <PolicySet> Permiso. El soporte de este tipo de <Policy> y, por tanto, de las peticiones sobre si un sujeto posee los privilegios asociados a determinado cometido, es facultativo.

Los ejemplares de un <PolicySet> Permiso han de almacenarse en un repositorio de políticas de modo que nunca se puedan utilizar como política inicial para un PDP en XACML. Los ejemplares del <PolicySet> Permiso deben estar disponibles únicamente a través del correspondiente <PolicySet> Cometido. Es necesario para soportar los cometidos jerárquicos, porque es condición que un <PolicySet> Permiso sea aplicable a todos los sujetos. El <PolicySet> Permiso confía en el <PolicySet> Cometido correspondiente para asegurar que sólo los sujetos que poseen el correspondiente atributo de cometido obtendrán acceso a los permisos de ese <PolicySet> Permiso.

El uso de ejemplares <PolicySet> Cometido y <PolicySet> Permiso separados permite utilizar el RBAC jerárquico, donde un cometido *mayor* puede adquirir los permisos de otro *menor*. Un <PolicySet> Permiso que no remita a otros elementos <PolicySet> Permiso podría ser una política y no un conjunto de políticas en XACML. Ahora bien, si se define como conjunto (<PolicySet>), el cometido asociado podrá formar parte de una jerarquía de cometidos ulteriormente, sin tener que cambiar las otras políticas.

8.1.4 Permisos de cometido múltiple

En este perfil es posible expresar políticas en las que un usuario debe poseer varios cometidos simultáneamente si pretende tener acceso a ciertos permisos. Por ejemplo, cambiar las instrucciones de cuidados a un paciente en un hospital puede precisar que el Subject que realice la acción desempeñe tanto un cometido médico como de empleado.

Estas políticas pueden expresarse mediante un <PolicySet> Cometido en el que el elemento <Target> precisa que el Subject debe tener todos los atributos de cometido necesarios. Esto se consigue usando un único elemento <Subject> que contiene múltiples elementos <SubjectMatch>. El <PolicySet> Permiso asociado debe especificar los permisos asociados a sujetos que tienen habilitados simultáneamente todos los cometidos especificados.

El <PolicySet> Permiso asociado a una política de cometido múltiple puede remitir a ejemplares de <PolicySet> Permiso asociadas a otros cometidos, pudiendo así heredar permisos de otros cometidos. Los permisos asociados a un determinado <PolicySet> de cometido múltiple también pueden ser heredados por otro cometido, si éste incluye una referencia al <PolicySet> Permiso asociado a la política de cometido múltiple en su propio <PolicySet> Permiso.

8.2 Ejemplo del control de acceso basado en cometidos

Esta cláusula es informativa.

Esta cláusula presenta un ejemplo completo de los tipos de políticas asociadas al control de acceso basado en el cometido.

Supóngase que una organización utiliza dos cometidos, de directivo y empleado. En este ejemplo, se expresan como dos valores separados para un solo Atributo XACML que tiene el AttributeId "&role;". Los valores de este identificador para estos cometidos son "&roles;employee" y "&roles;manager". Un empleado tiene permiso para crear una orden de compra. Un directivo tiene permiso para firmar una orden de compra, y todos los permisos que tiene el cometido de empleado. Por consiguiente, el cometido de directivo es mayor que el de empleado, y el cometido de empleado es menor que el de directivo.

En este perfil habrá dos ejemplares <PolicySet> Permiso: una para el cometido de directivo y otra para el de empleado. El <PolicySet> Permiso de directivo dará a cualquier sujeto el permiso específico de firmar una orden de compra y remitirá al <PolicySet> Permiso de empleado para heredar sus permisos. El <PolicySet> Permiso de empleado facilitará a cualquier Subject el permiso para crear una orden de compra.

En este perfil también habrá dos ejemplares de <PolicySet> Cometido: una para el cometido de directivo y otra para el de empleado. El <PolicySet> Cometido de directivo contendrá un <Target> que exige que el Subject tenga el identificador &role "&roles;manager". Remitirá al <PolicySet> Permiso de directivo. El <PolicySet> Cometido de empleado contendrá un <Target> que exige que el sujeto tenga el identificador &role "&roles;employee", y remitirá al <PolicySet> Permiso de empleado.

8.2.1 <PolicySet> Permiso para el cometido de directivo

El <PolicySet> Permiso siguiente contiene los permisos asociados con el cometido de directivo. En el PDP regirá un sistema de aplicación de políticas que sólo permite acceder a este <PolicySet> mediante referencia desde el <PolicySet> Cometido de directivo (véase el cuadro 8-1).

Cuadro 8-1/X.1142 – <PolicySet> Permiso para directivos

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
PolicySetId="PPS:manager:role"
PolicyCombiningAlgId="&policy-combine;permit-overrides">

<!-- Permissions specifically for the manager role -->
<Policy PolicyId="Permissions:specifically:for:the:manager:role"
    RuleCombiningAlgId="&rule-combine;permit-overrides">
    <!-- Permission to sign a purchase order -->
    <Rule RuleId="Permission:to:sign:a:purchase:order" Effect="Permit">
                <Target>
                      <Resources>
                            <Resource>
                                  <ResourceMatch MatchId="&function;string-equal">
                                     <AttributeValue
                                          DataType="&xml;string">purchase
order</AttributeValue>
                                  <ResourceAttributeDesignator
AttributeId="&resource;resource-id"
                                          DataType="&xml;string"/>
                            </ResourceMatch>
                            </Resource>
                      </Resources>
                      <Actions>
                            <Action>
                                  <ActionMatch MatchId="&function;string-equal">
                                    <AttributeValue
                                       DataType="&xml;string">sign</AttributeValue>
                                       ActionAttributeDesignator
                                       AttributeId="&action;action-id"
                                       DataType="&xml;string"/>
                                  </ActionMatch>
                            </Action>
                      </Actions>
                </Target>
          </Rule>
</Policy>
<!-- Include permissions associated with employee role -->
<PolicySetIdReference>PPS:employee:role</PolicySetIdReference>
</PolicySet>
```

8.2.2 <PolicySet> Permiso para el cometido de empleado

El siguiente <PolicySet> Permiso contiene los permisos asociados con el cometido de empleado (véase el cuadro 8-2). El sistema de aplicación de políticas en el PDP debe establecerse de modo que el acceso a este <PolicySet> se obtenga tan sólo mediante referencia desde el <PolicySet> Cometido de empleado, o desde el <PolicySet> Cometido superior de directivo, mediante el <PolicySet> Permiso de directivo.

Cuadro 8-2/X.1142 – <PolicySet> Permiso para empleados

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   PolicySetId="PPS:employee:role"
   PolicyCombiningAlgId="&policy-combine;permit-overrides">
   <!-- Permissions specifically for the employee role -->
   <Policy PolicyId="Permissions:specifically:for:the:employee:role"
          RuleCombiningAlgId="&rule-combine;permit-overrides">
   <!-- Permission to create a purchase order -->
          <Rule RuleId="Permission:to:create:a:purchase:order" Effect="Permit">
                 <Target>
                        <Resources>
                               <Resource>
                                      <ResourceMatch MatchId="&function;string-equal">
                                        <AttributeValue
                                        DataType="&xml;string">purchase
order</AttributeValue>
                                        <ResourceAttributeDesignator
AttributeId="&resource;resource-id"
                                        DataType="&xml;string"/>
                                      </ResourceMatch>
                               </Resource>
                        </Resources>
                        <Actions>
                               <Action>
                                      <ActionMatch MatchId="&function;string-equal">
                                        <AttributeValue
DataType="&xml;string">create</AttributeValue>
                                        <ActionAttributeDesignator
AttributeId="&action;action-id"
                                        DataType="&xml;string"/>
                                      </ActionMatch>
                               </Action>
                        </Actions>
                 </Target>
          </Rule>
   </Policy>
</PolicySet>
```

8.2.3 <PolicySet> Cometido para el cometido de directivo

El `<PolicySet>` Cometido siguiente es aplicable, según su `<Target>`, tan sólo a `Subjects` que tengan el identificador `&role` (véase el cuadro 8-3) "`&roles;manager`". La `<PolicySetIdReference>` indica el `<PolicySet>` Permiso asociado con el cometido de directivo.

Cuadro 8-3/X.1142 – <PolicySet> Cometido para directivos

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   PolicySetId="RPS:manager:role"
   PolicyCombiningAlgId="&policy-combine;permit-overrides">
   <Target>
          <Subjects>
                 <Subject>
                        <SubjectMatch MatchId="&function;anyURI-equal">
                               <AttributeValue
DataType="&xml;anyURI">&roles;manager</AttributeValue>
                               <SubjectAttributeDesignator AttributeId="&role;"
                                      DataType="&xml;anyURI"/>
                        </SubjectMatch>
                 </Subject>
          </Subjects>
   </Target>
<!-- Use permissions associated with the manager role -->
   <PolicySetIdReference>PPS:manager:role</PolicySetIdReference>
</PolicySet>
```

8.2.4 <PolicySet> Cometido para el cometido de empleado

El `<PolicySet>` Cometido siguiente es aplicable, según su `<Target>`, tan sólo a `Subjects` que tengan el identificador `&role` (véase el cuadro 8-4) `"&roles;employee"`. La `<PolicySetIdReference>` indica el `<PolicySet>` Permiso asociado con el cometido de empleado.

Cuadro 8-4/X.1142 – <PolicySet> Cometido para empleados

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   PolicySetId="RPS:employee:role"
   PolicyCombiningAlgId="&policy-combine;permit-overrides">
   <Target>
          <Subjects>
                 <Subject>
                        <SubjectMatch MatchId="&function;anyURI-equal">
                               <AttributeValue
DataType="&xml;anyURI">&roles;employee</AttributeValue>
                               <SubjectAttributeDesignator AttributeId="&role;"
DataType="&xml;anyURI"/>
                        </SubjectMatch>
                 </Subject>
          </Subjects>
   </Target>
<!-- Use permissions associated with the employee role -->
<PolicySetIdReference>PPS:employee:role</PolicySetIdReference>
</PolicySet>
```

8.2.5 Políticas y peticiones relativas a los privilegios de un Cometido

En un sistema de acceso RBAC XACML son posibles las consultas del tipo "¿Tiene este sujeto los privilegios del cometido X?" Para ello, cada `<PolicySet>` Permiso debe contener una `<Policy> HasPrivilegesOfRole`. En el caso del `<PolicySet>` Permiso para directivos, la `<Policy> HasPrivilegesOfRole` se escribiría como en el cuadro 8-5.

Cuadro 8-5/X.1142 – <PolicySet> Permiso para directivos

```
<!-- HasPrivilegesOfRole Policy for manager role -->
<Policy PolicyId="Permission:to:have:manager:role:permissions"
   RuleCombiningAlgId="&rule-combine;permit-overrides">
<!-- Permission to have manager role permissions -->
   <Rule RuleId="Permission:to:have:manager:permissions" Effect="Permit">
          <Condition>
                 <Apply FunctionId="&function;and">
                        <Apply FunctionId="&function;anyURI-is-in">
                               <AttributeValue
DataType="&xml;anyURI">&roles;manager</AttributeValue>
                               <ResourceAttributeDesignator AttributeId="&role;"
DataType="&xml;anyURI"/>
                        </Apply>
                               <Apply FunctionId="&function;anyURI-is-in">
                               <AttributeValue

   DataType="&xml;anyURI">&actions;hasPrivilegesofRole</AttributeValue>
                               <ActionAttributeDesignator AttributeId="&action;action-
id"
                                      DataType="&xml;anyURI"/>
                        </Apply>
                 </Apply>
          </Condition>
   </Rule>
</Policy>
```

En el caso del <PolicySet> Permiso para empleados, la <Policy> HasPrivilegesOfRole se escribiría como en el cuadro 8-6.

Cuadro 8-6/X.1142 – <PolicySet> Permiso para empleados

```
<!-- HasPrivilegesOfRole Policy for employee role -->
<Policy PolicyId="Permission:to:have:employee:role:permissions"
   RuleCombiningAlgId="&rule-combine;permit-overrides">
<!-- Permission to have employee role permissions -->
   <Rule RuleId="Permission:to:have:employee:permissions" Effect="Permit">
          <Condition>
                <Apply FunctionId="&function;and">
                      <Apply FunctionId="&function;anyURI-is-in">
                            <AttributeValue
DataType="&xml;anyURI">&roles;employee</AttributeValue>
                            <ResourceAttributeDesignator AttributeId="&role;"
                                  DataType="&xml;anyURI"/>
                      </Apply>
                      <Apply FunctionId="&function;anyURI-is-in">
                      <AttributeValue
DataType="&xml;anyURI">&actions;hasPrivilegesofRole
                      </AttributeValue>
                      <ActionAttributeDesignator AttributeId="&action;action-id"
                            DataType="&xml;anyURI"/>
                      </Apply>
                </Apply>
          </Condition>
   </Rule>
</Policy>
```

Una Petición que solicita si el sujeto Anne tiene los privilegios del atributo &roles;manager se escribiría como en el cuadro 8-7.

Cuadro 8-7/X.1142 – Petición acerca de un sujeto

```
<Request>
   <Subject>
          <Attribute AttributeId="&subject;subject-id" DataType="&xml;string">
                <AttributeValue>Anne</AttributeValue>
          </Attribute>
   </Subject>
   <Resource>
          <Attribute AttributeId="&role;" DataType="&xml;anyURI">
                <AttributeValue>&roles;manager</AttributeValue>
                </Attribute>
   </Resource>
   <Action>
          <Attribute AttributeId="&action;action-id"

   DataType="&xml;anyURI">&actions;hasPrivilegesOfRole</AttributeValue>
          </Attribute>
   </Action>
</Request>
```

Es necesario que la <Request> contenga los cometidos directos de Anne (en este caso, &roles;employee), o que el controlador de contexto del PDP pueda descubrirlos. Las políticas HasPrivilegesOfRole no realizan la asociación de cometidos a sujetos.

8.3 Asignación y habilitación de atributos de cometido

Esta cláusula es informativa.

La asignación de distintos atributos de cometido a usuarios y la habilitación de dichos atributos en una sesión no están dentro de las funciones del DPD en XACML. Hay que implementar una o varias entidades separadas, que son las autoridades de habilitación de cometidos, para realizar las citadas funciones. En este perfil se supone que la presencia en el contexto de Petición XACML de un atributo de cometido para un usuario dado (Subject) es una asignación válida en el momento en que se solicita la decisión de acceso.

Por tanto, ¿de dónde vienen los atributos de cometidos de un sujeto? ¿Cuáles son las características de estas autoridades de habilitación de cometidos? La respuesta depende de la implementación, pero se pueden sugerir algunas posibilidades.

En algunos casos, los atributos de cometidos pueden proceder de un servicio de gestión de identidades que mantiene información acerca de un usuario, incluyendo los cometidos asignados o habilitados del sujeto; el servicio de gestión de identidades actúa como autoridad de habilitación de cometidos, pudiendo almacenar atributos de cometidos estáticos en un directorio LDAP. El controlador del contexto del PDP puede recuperar allí esos atributos o bien este servicio puede responder a las peticiones de este controlador del PDP para conocer los atributos de cometido de un sujeto (las peticiones son solicitudes de atributo SAML).

Las autoridades de habilitación de cometidos pueden usar una `<Policy>` o un `<PolicySet>` de asignación de cometidos XACML para determinar si un sujeto está autorizado a poseer un atributo de cometido particular y conocer el valor habilitado. Una `<Policy>` o `<PolicySet>` Asignación de Cometido responde a la pregunta "¿Está el sujeto X autorizado a tener habilitado el cometido R_i?" Esto no responde a la pregunta siguiente: "¿Qué conjunto de cometidos puede tener habilitados el sujeto X?" La autoridad de habilitación de cometidos debe tener algún modo de saber para qué cometido o cometidos tiene que presentar una petición. Por ejemplo, la autoridad de habilitación de cometidos puede mantener una lista de todos los cometidos posibles y, cuando se le pregunte cuáles son los cometidos asociados a un sujeto, puede cursar una petición relativa a las políticas de asignación de cometidos para cada uno de ellos.

En este perfil, las políticas de asignación de cometidos son un conjunto diferente de los ejemplares de `<PolicySet>` Cometido y `<PolicySet>` Permiso utilizados para determinar los permisos de acceso asociados a cada cometido. Las políticas de asignación de cometidos sólo se utilizan cuando la Petición XACML procede de una autoridad de habilitación de cometidos. Esta separación se puede gestionar de varias formas, por ejemplo con diferentes PDP que tengan contenedores de políticas diferentes o que exijan que se incluya en los elementos `<Request>` para peticiones de habilitación de cometido un `<Subject>` de categoría "`&subject-category;role-enablement-authority`".

No se ha definido ninguna forma fija para la `<Policy>` de Asignación de Cometidos. El ejemplo siguiente (cuadro 8-8) ilustra una posible, con dos elementos `<Rule>` XACML. El primer `<Rule>` declara que `Anne`, `Seth` y `Yassir` están autorizados a tener el cometido "`&roles;employee`" habilitado entre las 9 de la mañana y las 5 de la tarde. El segundo `<Rule>` declara que `Steve` está autorizado a tener el cometido "`&roles;manager`" habilitado, sin restricciones de hora.

Cuadro 8-8/X.1142 – Ejemplo de asignación de cometidos

```
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    PolicyId="Role:Assignment:Policy"
    RuleCombiningAlgId="&rule-combine;permit-overrides">
<!-- Employee role requirements rule -->
    <Rule RuleId="employee:role:requirements" Effect="Permit">
        <Target>
            <Subjects>
                <Subject>
                    <SubjectMatch MatchId="&function;string-equal">
                        <AttributeValue DataType="&xml;string">Seth</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="&subject;subject-id"
                            DataType="&xml;string"/>
                    </SubjectMatch>
                </Subject>
                <Subject>
                    <SubjectMatch MatchId="&function;string-equal">
                        <AttributeValue DataType="&xml;string">Anne</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="&subject;subject-id"
                            DataType="&xml;string"/>
                    </SubjectMatch>
                </Subject>
            </Subjects>
        <Resources>
            <Resource>
                <ResourceMatch MatchId="&function;anyURI-equal">
                    <AttributeValue
DataType="&xml;anyURI">&roles;employee</AttributeValue>
                        <ResourceAttributeDesignator AttributeId="&role;"
                            DataType="&xml;anyURI"/>
                    </ResourceMatch>
                    </Resource>
                </Resources>
```

Cuadro 8-8/X.1142 – Ejemplo de asignación de cometidos

```
                <Actions>
                <Action>
                    <ActionMatch MatchId="&function;anyURI-equal">
                        <AttributeValue DataType="&xml;anyURI">&actions;
                            enableRole</AttributeValue>
                        <ActionAttributeDesignator AttributeId="&action;action-id"
                            DataType="&xml;anyURI"/>
                    </ActionMatch>
                </Action>
            </Actions>
        </Target>
        <Condition>
            <Apply FunctionId="&function;and">
                <Apply FunctionId="&function;time-greater-than-or-equal">
                    <Apply FunctionId="&function;time-one-and-only">
                        <EnvironmentAttributeDesignator
AttributeId="&environment;current-time"
                            DataType="&xml;time"/>
                    </Apply>
                    <AttributeValue DataType="&xml;time">9h</AttributeValue>
                </Apply>
                <Apply FunctionId="&function;time-less-than-or-equal">
                    <Apply FunctionId="&function;time-one-and-only">
                        <EnvironmentAttributeDesignator
                AttributeId="&environment;current-time" DataType="&xml;time"/>
                    </Apply>
                    <AttributeValueDataType="&xml;time">17h</AttributeValue>
                </Apply>
        </Apply>
    </Condition>
</Rule>
<!-- Manager role requirements rule -->
    <Rule RuleId="manager:role:requirements" Effect="Permit">
        <Target>
            <Subjects>
                <Subject>
                    <SubjectMatch MatchId="&function;string-equal">
                        <AttributeValue DataType="&xml;string">Steve</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="&subject;subject-id"
                            DataType="&xml;string"/>
                    </SubjectMatch>
                </Subject>
            </Subjects>
            <Resources>
                <Resource>
                    <ResourceMatch MatchId="&function;anyURI-equal">
                        <AttributeValue
                            DataType="&xml;anyURI">&roles;:manager</AttributeValue>
                        <ResourceAttributeDesignator AttributeId="&role;"
                            DataType="&xml;anyURI"/>
                    </ResourceMatch>
                </Resource>
            </Resources>
        <Actions>
            <Action>
                <ActionMatch MatchId="&function;anyURI-equal">
                    <AttributeValue
                        DataType="&xml;anyURI">&actions;enableRole</AttributeValue>
                        <ActionAttributeDesignator AttributeId="&action;action-id"
                            DataType="&xml;anyURI"/>
                </ActionMatch>
            </Action>
        </Actions>
    </Target>
</Rule>
</Policy>
```

8.4 Implementación del modelo de acceso RBAC

Esta cláusula es informativa.

Las cláusulas siguientes describen cómo hay que utilizar las políticas XACML para implementar los distintos componentes del modelo de RBAC (véase [RBAC]).

8.4.1 RBAC fundamental

El RBAC fundamental incluye los siguientes cinco elementos de datos básicos:

- Los usuarios se implementan por medio de `Subjects` XACML. Puede utilizarse cualquiera de los valores de `SubjectCategory` XACML, según corresponda.
- Los cometidos se expresan mediante uno o varios atributos de sujeto XACML. El conjunto de cometidos es muy específico en función de la aplicación y del dominio de políticas, siendo muy importante que no se confundan los diferentes usos de cometidos. Por ello, este perfil no intenta definir ningún conjunto estándar de valores de cometido, aunque sí recomienda la utilización del valor común de `AttributeId` "`urn:oasis:names:tc:xacml:2.0:subject:role`". Se recomienda concertar y publicar en cada dominio de aplicación o de política un conjunto único de valores `AttributeId`, valores `DataType`, y valores `<AttributeValue>` que se utilizarán para los diversos cometidos pertinentes en dicho dominio.
- Los objetos se expresan mediante `Resources` XACML.
- Las operaciones se expresan mediante `Actions` XACML.
- Los permisos se expresan mediante ejemplares de `<PolicySet>` Cometido y `<PolicySet>` Permiso en XACML como se describe en las cláusulas anteriores.

El RBAC fundamental requiere que el sistema soporte múltiples usuarios por cometido, múltiples cometidos por usuario, múltiples permisos por cometido y múltiples cometidos por permiso. Cada uno de estos requisitos se puede satisfacer con las siguientes políticas XACML basadas en este perfil. Sin embargo, se señala que la asignación efectiva de cometidos a usuarios está fuera del alcance del PDP XACML.

El lenguaje XACML permite asociar múltiples sujetos a un determinado atributo de cometido. Los `<PolicySet>` Cometido de XACML definidos en términos de posesión de un determinado `<Attribute>` y `<AttributeValue>` de Cometido se aplicarán a cualquier usuario solicitante que tenga este `<Attribute>` y `<AttributeValue>` de Cometido en el contexto de petición XACML.

XACML permite asociar asimismo múltiples atributos de cometido o valores de atributos de cometido con un determinado `Subject`. Si un `Subject` tiene múltiples cometidos habilitados, entonces se puede evaluar cualquier ejemplar de un `<PolicySet>` Cometido que se aplique a cualquiera de dichos cometidos, y se concederán los permisos en el correspondiente `<PolicySet>` Permiso. Incluso es posible definir políticas que requieran que un `Subject` dado tenga múltiples atributos o valores de cometido habilitados simultáneamente. En tal caso, los permisos asociados con el requisito de cometido múltiple sólo valen para un `Subject` que tenga todos los valores y atributos de cometido necesarios cuando se presenta un contexto de Petición XACML al PDP para su evaluación.

El `<PolicySet>` Permiso asociado a un determinado cometido puede permitir el acceso a múltiples recursos mediante múltiples acciones. XACML tiene muchas composiciones de código para constituir permisos, de manera que hay múltiples formas en que se pueden expresar cometidos multi-permiso. Cualquier Cometido A se puede asociar con un `<PolicySet>` Permiso B incluyendo una referencia (`<PolicySetIdReference>`) a este conjunto de políticas B en el `<PolicySet>` Permiso asociado al Cometido A. De esta forma, el mismo conjunto de permisos se puede asociar a más de un cometido.

Además de los requerimientos básicos del RBAC fundamental, las políticas XACML que utilizan este perfil también pueden expresar condiciones arbitrarias sobre la aplicación de permisos particulares asociados a un cometido. Por ejemplo la limitación de los permisos a ciertas horas del día, o la limitación de los permisos a los poseedores del cometido que también poseen otros atributos, sean o no del mismo cometido.

8.4.2 RBAC jerárquico

El RBAC jerárquico añade al RBAC fundamental la capacidad de definir relaciones de herencia entre cometidos. Por ejemplo, el cometido A puede definirse para heredar todos los permisos asociados con el cometido B. En este caso, el cometido A se considera superior al cometido B en la jerarquía de cometidos. Si el cometido B a su vez hereda permisos asociados con el cometido C, entonces el cometido A también heredará dichos permisos por ser superior al cometido B.

Las políticas XACML que usan este perfil pueden implementar la herencia de cometidos incluyendo una <PolicySetIdReference> al <PolicySet> Permiso asociado con un cometido dentro del <PolicySet> Permiso asociado con otro cometido. El cometido que incluye la <PolicySetIdReference> heredará entonces los permisos asociados con el cometido referenciado.

En este perfil las políticas se estructuran de tal forma que las propiedades de herencia se pueden añadir a un cometido en cualquier momento sin modificar ejemplares del <PolicySet> asociadas con cualquier otro cometido. Es posible que una organización no utilice jerarquías de cometidos inicialmente, pero podrá hacerlo posteriormente sin tener que reescribir las políticas existentes.

8.5 Perfil

En esta cláusula se presentan los cometidos, los atributos de cometidos, la asignación de cometidos y el control de acceso.

8.5.1 Cometidos y atributos de cometidos

Los cometidos se deben expresar mediante uno o varios Atributos XACML. Cada dominio de aplicación que utilice este perfil para el control de acceso basado en cometidos deberá definir o acordar uno o varios valores AttributeId para usarlos como atributos de cometidos. Cada uno de estos valores AttributeId debe estar asociado a un conjunto de valores permitidos y sus DataTypes. Cada valor permitido para dicho AttributeId debe tener una semántica bien definida para utilizar el correspondiente valor en las políticas.

Este perfil recomienda el uso del valor AttributeId "urn:oasis:names:tc:xacml:2.0:subject:role" para todos los atributos de cometido. Los ejemplares de este atributo deben tener un DataType "http://www.w3.org/2001/XMLSchema#anyURI".

8.5.2 Habilitación o asignación de cometidos

Una Autoridad de Habilitación de Cometidos, responsable de la asignación de cometidos a usuarios y de la habilitación de cometidos para el uso dentro de una sesión de usuario, puede usar una <Policy> o un <PolicySet> Asignación de Cometidos de XACML para determinar qué usuarios están autorizados a habilitar qué cometidos y bajo qué condiciones. No hay una forma prescrita de <Policy> o <PolicySet>Asignación de Cometidos. Se recomienda que los cometidos en una <Policy> o <PolicySet> Asignación de Cometidos se expresen como Atributos de Recursos, utilizándose como identificador (AttributeId) "&role" y como <AttributeValue> el URI para el valor de cometido pertinentes. Se recomienda que la acción de asignar o habilitar un cometido se exprese como un atributo utilizándose como identificador (AttributeId) "&action;action-id", el DataType es &xml;anyURI, y el <AttributeValue> es "&actions;enableRole".

8.5.3 Control de acceso

El control de acceso basado en cometidos se debe implementar mediante dos tipos de <PolicySet>: el <PolicySet> Cometido y el <PolicySet> Permiso. Las funciones y los requisitos de estos dos tipos de conjuntos de políticas son:

Para cada cometido se debe definir un <PolicySet> Cometido. Dicho <PolicySet> debe contener un elemento <Target> que hace aplicable el <PolicySet> sólo a sujetos que tienen el atributo XACML asociado con ese cometido. El elemento <Target> no debe restringir el Resource, Action o Environment. Cada <PolicySet> Cometido debe contener un solo elemento <PolicySetIdReference> que remita al único <PolicySet> Permiso asociado con ese cometido. El <PolicySet> Cometido no debe contener ningún otro elemento <Policy>, <PolicySet>, <PolicyIdReference>, o <PolicySetIdReference>.

Para cada cometido se definirá un <PolicySet> Permiso que debe contener elementos <Policy> y <Rule> que especifiquen los tipos de acceso permitidos a los Subjects que tengan el cometido indicado. El <Target> del <PolicySet> y sus elementos <PolicySet>, <Policy>, y <Rule> incluidos o referenciados no deben limitar los Subjects a los que se aplica el <PolicySet> Permiso.

Si un cometido dado hereda permisos de uno o varios cometidos inferiores, entonces el <PolicySet> Permiso para el cometido (superior) dado debe incluir un elemento <PolicySetIdReference> para cada cometido inferior. Cada <PolicySetIdReference> debe remitir al <PolicySet> Permiso asociado con el cometido inferior del cual hereda el cometido superior.

Un <PolicySet> Permiso puede incluir una <Policy> HasPrivilegesOfRole. Esta política de privilegios del Cometido debe tener un elemento <Rule> con el efecto "Permitir". Esta regla debe permitir a cualquier sujeto realizar una acción con un atributo que tenga un identificador (AttributeId) "&action;action-id", un DataType "&xml;anyURI", y un valor (<AttributeValue>) "&actions;hasPrivilegesOfRole", para un recurso cuyo atributo sea el cometido al que aplica el <PolicySet> Permiso (por ejemplo, un AttributeId "&role", un

DataType "&xml;anyURI", y un <AttributeValue> que sea el URI del valor de cometido específico). Obsérvese que el atributo de cometido, que es un atributo de sujeto en un <Target> de un <PolicySet> Cometido, se trata como un atributo de recurso en una política de privilegios del Cometido (HasPrivilegesOfRole).

La organización de cualquier almacén utilizado para políticas y la configuración del PDP deben impedir que el PDP utilice un <PolicySet> Permiso como política inicial del PDP.

8.6 Identificadores

Este perfil define los siguientes identificadores URN.

8.6.1 Identificador de perfil

El siguiente identificador se debe usar para este perfil cuando se requiere un identificador en la forma de un URI.

```
urn:oasis:names:tc:xacml:2.0:profiles:rbac:core-hierarchical
```

8.6.2 Atributo de cometido

El siguiente identificador se puede usar como AttributeId para atributos de cometido.

```
urn:oasis:names:tc:xacml:2.0:subject:role
```

8.6.3 Categoría del Sujeto

El siguiente identificador se puede usar como SubjectCategory para atributos de sujeto que señalan que una petición proviene de una autoridad de habilitación de cometidos.

```
urn:oasis:names:tc:xacml:2.0:subject-category:role-enablement-authority
```

8.6.4 Valores de atributo de acción

El siguiente identificador se puede usar como <AttributeValue> del atributo "&action;action-id" en HasPrivilegesOfRole <Policy>.

```
urn:oasis:names:tc:xacml:2.0:actions:hasPrivilegesOfRole
```

El siguiente identificador se puede utilizar como <AttributeValue> del atributo "&action;action-id" en una política de asignación de cometidos.

```
urn:oasis:names:tc:xacml:2.0:actions:enableRole
```

9 Perfil de recurso múltiple de XACML

La evaluación de política realizada por un punto de decisión de política XACML, o PDP, se define como un solo recurso solicitado, con una decisión de autorización que se comunica en un elemento <Result> del contexto de respuesta. Sin embargo, en algunos casos puede ser conveniente que un punto de cumplimiento de política, o PEP, presente un único contexto de petición para acceder a múltiples recursos, así como obtener un único contexto de respuesta que contenga una decisión de autorización separada (elemento <Result>) para cada recurso solicitado. Dicho contexto de petición se puede usar para evitar el envío de múltiples mensajes de petición de decisión entre el PEP y el PDP, por ejemplo. Asimismo, puede ser conveniente que un PEP presente un único contexto de petición para todos los nodos de una jerarquía, y obtener una única decisión de autorización (elemento <Result>) que indique si se permite el acceso a todos los nodos solicitados. Dicho contexto de petición se puede usar cuando el solicitante desea acceder a todo un documento XML, a todo un sub-árbol de elementos en dicho documento, o a todo un directorio de sistemas de archivos con todos sus subdirectorios y archivos, por ejemplo.

Esta Recomendación describe tres opciones para la petición de decisiones de autorización del PEP para múltiples recursos en un único contexto de petición, y cómo se representa el resultado de cada decisión de autorización en el contexto de respuesta único que se devuelve al PEP.

Esta Recomendación también describe dos opciones de petición del PEP para recibir una sola decisión de autorización como respuesta a una petición para todos los nodos de una jerarquía.

La posibilidad de utilizar los mecanismos descritos en este perfil es facultativa para las implementaciones XACML que se ajustan a la normativa.

Los atributos de recurso usados normalmente tienen las siguientes abreviaturas:

– Atributo "resource-id" – un atributo de recurso cuyo AttributeId es:

  "urn:oasis:names:tc:xacml:1.0:resource:resource-id".

– Atributo "scope" – un atributo de recurso cuyo AttributeId es:

  "urn:oasis:names:tc:xacml:2.0:resource:scope".

  Véase 9.3 para ampliar información sobre este atributo.

9.1 Peticiones de recursos múltiples

Esta cláusula es normativa, aunque facultativa.

Un único contexto de Petición XACML puede representar una petición de acceso a recursos múltiples, y se espera una decisión de autorización separada para cada recurso. La sintaxis y semántica de dichas peticiones y respuestas se especifican en esta cláusula.

Los elementos <Result> producidos al evaluar una petición de acceso a recursos múltiples deben ser idénticos a los que se producirían por una serie de peticiones, cada una solicitando acceso a uno solo de los recursos (un recurso individual). El contexto de petición conceptual que corresponde a cada elemento <Result> se denomina "petición de recurso individual". El valor ResourceId en el elemento <Result> debe ser el <AttributeValue> del atributo "resource-id" en la correspondiente petición de recurso individual. Esta operación de reflejar un contexto de petición original que contiene múltiples peticiones de decisión, en peticiones de recurso individuales, y reflejar múltiples decisiones de autorización en múltiples elementos <Result> en un único contexto de respuesta, puede hacerla el controlador o función de servicio de contexto en esta Recomendación. Este perfil no requiere que se construya la implementación de la evaluación de una petición de acceso a recursos múltiples conforme al modelo descrito, ni que se construyan peticiones de recurso individuales. El perfil requiere solamente que los elementos <Result> sean los mismos que se habrían obtenido con este modelo.

En las cláusulas siguientes se describen tres formas de especificar peticiones de acceso a recursos múltiples. Cada una de ellas detalla las peticiones de recurso individual que corresponden a los elementos <Result> en el contexto de respuesta.

Un único contexto Petición XACML presentado por un PEP puede usar más de una de estas formas de petición de acceso a recursos múltiples en elementos <Resource> diferentes.

9.1.1 Nodos identificados mediante el atributo "scope"

Esta cláusula es normativa, aunque facultativa.

En este texto se describe el uso de dos valores del atributo de recurso "scope" para especificar una petición de acceso a recursos múltiples en una jerarquía. Esta sintaxis se puede usar con cualquier recurso jerárquico, sea o no un documento XML.

9.1.1.1 URI del Perfil

Los siguientes URI se deben usar como identificadores URI para la funcionalidad especificada en esta parte de este perfil. El primer identificador se debe usar cuando la funcionalidad es soportada por recursos XML, y el segundo cuando la funcionalidad es soportada por recursos que no son documentos XML:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:scope:xml
urn:oasis:names:tc:xacml:2.0:profile:multiple:scope:non-xml
```

9.1.1.2 Sintaxis del contexto de petición original

El elemento <Resource> del contexto de Petición XACML original debe contener un atributo "scope" con un valor "Children" (vástagos) o "Descendants" (descendientes). Si los recursos solicitados están en un documento XML, entonces en el elemento <ResourceContent> se incluirá todo el documento XML en el que se encuentran los elementos solicitados. Asimismo, si los recursos solicitados están en un documento XML, el resultado de evaluación de la expresión XPath usada como valor del atributo "resource-id" ha de ser un conjunto de nodos que sólo contiene un nodo.

9.1.1.3 Semántica

Dicho contexto de petición se debe interpretar como una petición de acceso a un conjunto de nodos en una jerarquía relativa al nodo único especificado en el atributo "resource-id". Si el valor del atributo "scope" es "Children", cada recurso individual es el nodo único indicado por el atributo "resource-id" (pueden ser varios atributos si el recurso

único tiene múltiples identificadores normativos) y todos sus nodos vástagos inmediatos. Si el valor del atributo "scope" es "Descendants", el recurso individual es el nodo único indicado por el atributo "resource-id" y todos sus nodos descendientes.

Cada petición de recurso individual debe ser idéntica al contexto de petición original con dos excepciones: el atributo "scope" no debe estar presente y el elemento <Resource> debe representar un recurso individual único. Este elemento <Resource> debe contener al menos un atributo "resource-id", y todos los valores de dichos atributos deben ser identidades normativas únicas del recurso individual. Si el atributo "resource-id" en el contexto de petición original contenía un emisor, los atributos "resource-id" en la petición de recurso individual deben contener el mismo emisor. Si en el contexto de petición original había un elemento <ResourceContent>, entonces el mismo elemento <ResourceContent> debe estar incluido en cada petición de recurso individual.

Ni XACML ni este perfil especifican cómo la función de servicio de contexto obtiene la información requerida para determinar qué nodos son vástagos o descendientes de un nodo dado, excepto en el caso de un documento XML, en que la información se debe obtener a partir del elemento <ResourceContent>.

9.1.2 Nodos identificados mediante XPath

Esta cláusula es normativa, aunque facultativa.

En esta subcláusula se describe el uso de una expresión XPath en el atributo "resource-id", junto con un valor de "XPath-expression" en el atributo "scope" para especificar una petición de acceso a nodos múltiples en un documento XML. Esta sintaxis se debe usar sólo con recursos que sean documentos XML.

9.1.2.1 URI del perfil

Se utilizará el siguiente URI como URI identificador para la funcionalidad especificada en esta parte de este perfil:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:xpath-expression
```

9.1.2.2 Contexto de petición original

El elemento <Resource> del contexto de Petición XACML original debe contener un elemento <ResourceContent> y un atributo "resource-id" con un DataType de "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". El <AttributeValue> del atributo "resource-id" será una expresión XPath que produzca como resultado de evaluación un conjunto de nodos que representan los múltiples nodos del elemento <ResourceContent>. El elemento <Resource> debe contener un atributo "scope" con un valor de "XPath-expression".

9.1.2.3 Semántica

Dicho contexto de petición se debe interpretar como una petición de acceso a los nodos del conjunto representado por el <AttributeValue> del atributo "resource-id". Cada uno de dichos nodos debe representar un recurso individual.

Cada petición de recurso individual debe ser idéntica al contexto de petición original con dos excepciones: el atributo "scope" no debe estar presente y el atributo "resource-id" debe ser una expresión XPath que produzca como resultado de evaluación un único nodo en el elemento <ResourceContent>. Ese nodo debe ser el recurso individual. Si el atributo "resource-id" en el contexto de petición original contenía un emisor, el atributo "resource-id" en la petición de recurso individual debe contener el mismo emisor.

9.1.3 Múltiples elementos <Resource>

Esta cláusula es normativa, aunque facultativa.

En esta cláusula se describe el uso de múltiples elementos <Resource> en un contexto de petición para especificar peticiones de acceso a recursos múltiples. Esta sintaxis se puede usar con cualquier recurso o recursos, sin tener en cuenta si son documentos XML o recursos jerárquicos.

9.1.3.1 URI del Perfil

Se utilizará el siguiente URI como URI identificador para la funcionalidad especificada en esta parte de este perfil:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:multiple-resource-elements
```

9.1.3.2 Contexto de petición original

El contexto de Petición XACML debe contener múltiples elementos <Resource>.

9.1.3.3 Semántica

Dicho contexto de petición se debe interpretar como una petición de acceso a todos los recursos especificados en los elementos <Resource> individuales. Cada elemento <Resource> debe representar un recurso individual a menos que el elemento utilice los otros mecanismos descritos en este perfil.

Para cada elemento <Resource> se debe crear una petición de recurso individual. Dicha petición debe ser idéntica al contexto de petición original con una excepción: sólo debe estar presente el elemento único <Resource>. Si dicho elemento <Resource> contiene un atributo "scope" con un valor distinto de "Immediate", entonces la petición de recurso individual se debe seguir procesando según la parte correspondiente de este perfil. Dicho procesamiento puede implicar la descomposición de la petición de recurso individual único en otras peticiones de recursos individuales antes de la evaluación que realiza el PDP.

9.2 Peticiones para toda la jerarquía

Esta cláusula es normativa, aunque facultativa.

En algunos casos, para un recurso jerárquico se emite una petición de decisión de autorización para acceder a todos los nodos dentro de dicho recurso, o a una subjerarquía entera de nodos dentro del mismo. Por ejemplo cuando se solicita el acceso a un documento XML con el propósito de hacer una copia del documento entero, o se solicita acceso a un directorio entero de archivos con todos sus subdirectorios y archivos. Se desea un único <Result>, indicando si el solicitante puede acceder a todo el conjunto de nodos.

El elemento <Result> producido al evaluar dicha petición de acceso debe ser idéntico al producido por el proceso siguiente. Se evalúa una serie de contextos de petición, cada uno solicitando acceso exactamente a un nodo de la jerarquía. La <Decision> en el único <Result> que se devuelve al PEP será "Permitir" sólo si todos los elementos <Result> resultantes de la evaluación de los nodos individuales contenían una <Decision> de "Permitir". De otro modo, la <Decision> en el único <Result> que se devuelve al PEP debe ser "Denegar". Este perfil no requiere esta forma de evaluación de una petición de acceso a dichos recursos jerárquicos, ni que se creen efectivamente contextos de petición correspondientes a los nodos individuales en la jerarquía; el único requisito es que los elementos <Result> sean los mismos que si se usara el modelo precedente.

En las cláusulas siguientes se especifican dos sintaxis para dicha funcionalidad, una para el uso con recursos que son documentos XML y otra para el uso con recursos que no son documentos XML.

9.2.1 Recursos XML

Esta cláusula es normativa, aunque facultativa.

En esta cláusula se describe la sintaxis para solicitar acceso a todo un documento XML, o a un elemento dentro de ese documento con todos sus subelementos recursivos.

9.2.1.1 URI del Perfil

Se utilizará el siguiente URI como identificador para la funcionalidad especificada en esta parte de este perfil:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:entire-hierarchy:xml
```

9.2.1.2 Contexto de petición original

El elemento <Resource> del contexto de petición original debe contener un atributo "scope" con un valor "EntireHierarchy".

El elemento <Resource> del contexto de petición original debe contener un atributo "resource-id" único con un DataType "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". El resultado de evaluación del <AttributeValue> ha de ser un conjunto de nodos que representa exactamente un nodo del elemento <ResourceContent>.

El elemento <Resource> del contexto de petición original puede contener otros atributos.

9.2.1.3 Semántica

El elemento <Result> de dicha petición debe ser equivalente al producido por el proceso siguiente. Para cada nodo en la jerarquía solicitada, la función de servicio del contexto debe crear un nuevo contexto de petición. Cada contexto de petición debe contener un único elemento <Resource> que tenga un atributo "resource-id" con un DataType "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression"; su valor ha de ser una expresión XPath que produzca como resultado de evaluación un conjunto de nodos que contiene exactamente ese nodo del elemento <ResourceContent>. La función de servicio del contexto debe presentar cada nuevo contexto de petición al PDP para

su evaluación y tomará nota de la <Decision> en los correspondientes elementos <Result>. Si el resultado de evaluación de todos los nuevos contextos de petición es "Permitir" (sólo en este caso), habrá un único <Result> con una <Decision> "Permitir" en el contexto de respuesta devuelto al PEP. Si el resultado de evaluación de uno de los nuevos contextos de petición fuera "Denegar", "Indeterminate", o "NotApplicable", habrá un único <Result> con una <Decision> "Denegar" en el contexto de respuesta que se devuelve al PEP.

9.2.2 Recursos no-XML

Esta cláusula es normativa, aunque facultativa.

En esta cláusula se describe la sintaxis para solicitar acceso a una jerarquía entera de nodos dentro de un recurso jerárquico que no es un documento XML.

9.2.2.1 URI del Perfil

Se utilizará el siguiente URI como identificador para la funcionalidad especificada en esta parte de este perfil:

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:entire-hierarchy:non-xml
```

9.2.2.2 Contexto de petición original

El elemento <Resource> en el contexto de petición original debe contener un atributo "scope" con un valor "EntireHierarchy".

El elemento <Resource> en el contexto de petición original debe contener un atributo "resource-id" único que represente un único nodo en un recurso jerárquico.

El elemento <Resource> en el contexto de petición original puede contener otros atributos.

La representación de nodos en un recurso jerárquico especificado en el perfil XACML para recursos jerárquicos en esta Recomendación se puede usar para representar la identidad del nodo único.

9.2.2.3 Semántica

El <Result> de dicha petición debe ser equivalente al producido por el proceso siguiente. Para cada nodo en la jerarquía solicitada, la función de servicio del contexto debe crear un nuevo contexto de petición. Cada contexto de petición debe contener un único elemento <Resource> que tenga un atributo "resource-id" con un valor que es la identidad del único nodo de la jerarquía. La función de servicio del contexto debe presentar cada nuevo contexto de petición al PDP para su evaluación, y tomará nota de la <Decision> en los correspondientes elementos <Result>. Si el resultado de evaluación de todos los nuevos contextos de petición es "Permitir" (sólo en este caso), habrá un único <Result> con una <Decision> "Permitir" en el contexto de respuesta que se devuelve al PEP. Si el resultado de uno de los nuevos contextos de petición fuera "Denegar", "Indeterminate", o "NotApplicable", habrá un único <Result> con una <Decision> "Denegar" en el contexto de respuesta que se devuelve al PEP.

Ni XACML ni este perfil especifican cómo la función de servicio de contexto obtiene la información solicitada para determinar qué nodos son descendientes del nodo especificado originalmente, o cómo se representa la identidad de cada nodo. La representación de nodos en un recurso jerárquico especificado en el perfil XACML para recursos jerárquicos en esta Recomendación puede utilizarse para representar la identidad de cada nodo.

9.3 Nuevos identificadores de atributos

El siguiente identificador se utiliza como AttributeId de un atributo de recursos que indica el alcance (atributo "scope") de una petición de acceso en un único elemento <Resource> de un contexto de petición.

```
urn:oasis:names:tc:xacml:2.0:resource:scope
```

El atributo debe tener el DataType "http://www.w3.org/2001/XMLSchema#string".

Los valores válidos para este atributo se indican a continuación, así como en 7.5. Una implementación puede soportar cualquier subconjunto de estos valores, incluyendo el conjunto vacío.

- "Immediate" – El elemento <Resource> se refiere a un único recurso no jerárquico o a un único nodo en un recurso jerárquico. Éste es el valor por defecto si no se ha asignado ningún atributo "scope". El elemento <Resource> se debe procesar como se indica en la cláusula 7.
- "Children" – El elemento <Resource> se refiere a múltiples recursos en una jerarquía. El conjunto de recursos consiste en un único nodo descrito por el atributo de recurso "resource-id" y todos los

vástagos inmediatos del nodo en la jerarquía. El elemento <Resource> se debe procesar como se indica en 9.1.1 de este perfil.

– "Descendants" – El elemento <Resource> se refiere a múltiples recursos en una jerarquía. El conjunto de recursos consiste en un único nodo descrito por el atributo de recurso "resource-id" y en todos los descendientes de ese nodo en la jerarquía. El elemento <Resource> se debe procesar como se indica en 9.1.1 de este perfil.

– "XPath-expression" – El elemento <Resource> se refiere a múltiples recursos. El conjunto de recursos consiste en los nodos de un conjunto descrito por el atributo de recurso "resource-id". Cada uno de estos nodos debe formar parte del elemento <ResourceContent> del elemento <Resource>. El elemento <Resource> se debe procesar como se indica en 9.1.2 de este perfil.

– "EntireHierarchy" – El elemento <Resource> se refiere a un único recurso. El recurso consiste en un nodo descrito por el atributo de recurso "resource-id" y todos los descendientes de ese nodo. Todos los nodos deben estar incluidos en un documento XML que forma parte del elemento <ResourceContent> del elemento <Resource>. El elemento <Resource> se debe procesar como se indica en 9.2.

9.4 Nuevos identificadores de perfiles

Los siguientes valores de URI se deben usar como identificadores URI para la funcionalidad especificada en distintas secciones de este perfil:

– Los atributos "children" o "descendants" "scope" en <Resource>: recursos XML

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:scope:xml
```

– Los atributos "children" o "descendants" "scope" en <Resource>: recursos no-XML

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:scope:non-xml
```

– Expresión XPath en atributo "resource-id"

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:xpath-expression
```

– Múltiples elementos <Resource>

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:multiple-resource-elements
```

– Peticiones para toda la jerarquía: recursos XML

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:entire-hierarchy:xml
```

– Peticiones para toda la jerarquía: recursos no-XML

```
urn:oasis:names:tc:xacml:2.0:profile:multiple:entire-hierarchy:non-xml
```

10 Perfil SAML 2.0 de XACML

En esta cláusula se define un perfil relativo a la utilización de SAML 2.0 (véase la Rec. UIT-T X.1141) para proteger, transportar, y solicitar instancias de un esquema XACML y otras informaciones necesarias para una implementación XACML.

SAML es un marco de trabajo basado en XML para el intercambio de información sobre seguridad, que se expresa en forma de aserciones acerca de sujetos. El sujeto es una entidad (una persona o un ordenador) que posee una identidad en algún dominio de seguridad. Una sola aserción puede contener varias declaraciones internas diferentes acerca de autenticación, autorización y atributos. SAML define un protocolo para las solicitudes de aserciones de los clientes a las autoridades SAML y para las respuestas de éstas. Este protocolo, consistente en formatos de mensajes de petición y respuesta basado en XML, puede integrarse en muchas comunicaciones subyacentes y muchos protocolos de transporte diferentes; SAML define actualmente una forma de integración a SOAP mediante HTTP. Para crear sus respuestas, las autoridades SAML pueden usar varias fuentes de información, por ejemplo registros de políticas externos y aserciones que se reciben como entradas en las peticiones. SAML define elementos, sujetos, condiciones, avisos y declaraciones de aserción.

Hay seis tipos de consultas y aseveraciones usadas en esta cláusula:

1) `AttributeQuery`: Una petición SAML normal utilizada para solicitar uno o varios atributos a una autoridad de atributos.
2) `AttributeStatement`: Una declaración SAML normal que contiene uno o varios atributos. Esta declaración se puede usar en una respuesta SAML de una autoridad de atributos, o se puede usar en una aserción SAML como formato para guardar atributos en un registro.
3) `XACMLPolicyQuery`: Una extensión de petición SAML, definida en este perfil. Se usa para pedir una o varias políticas a un punto de administración de políticas.
4) `XACMLPolicyStatement`: Una extensión de declaración SAML definida en este perfil. Se puede usar en una respuesta SAML de un punto de administración de políticas, y también en una aserción SAML como formato para guardar políticas en un registro.
5) `XACMLAuthzDecisionQuery`: Una extensión de petición SAML definida en este perfil. La utiliza el PEP para solicitar una decisión de autorización a un PDP XACML.
6) `XACMLAuthzDecisionStatement`: Una extensión de declaración SAML definida en este perfil. Se puede usar en una respuesta SAML de un PDP XACML. También se puede usar en una aserción SAML que sirve de credencial, pero no es parte del modelo de uso XACML definido actualmente.

El siguiente diagrama (figura 10-1) ilustra el modelo de uso XACML y los mensajes que se utilizan para comunicar entre los distintos componentes. No todos los componentes se usarán en cada implementación.

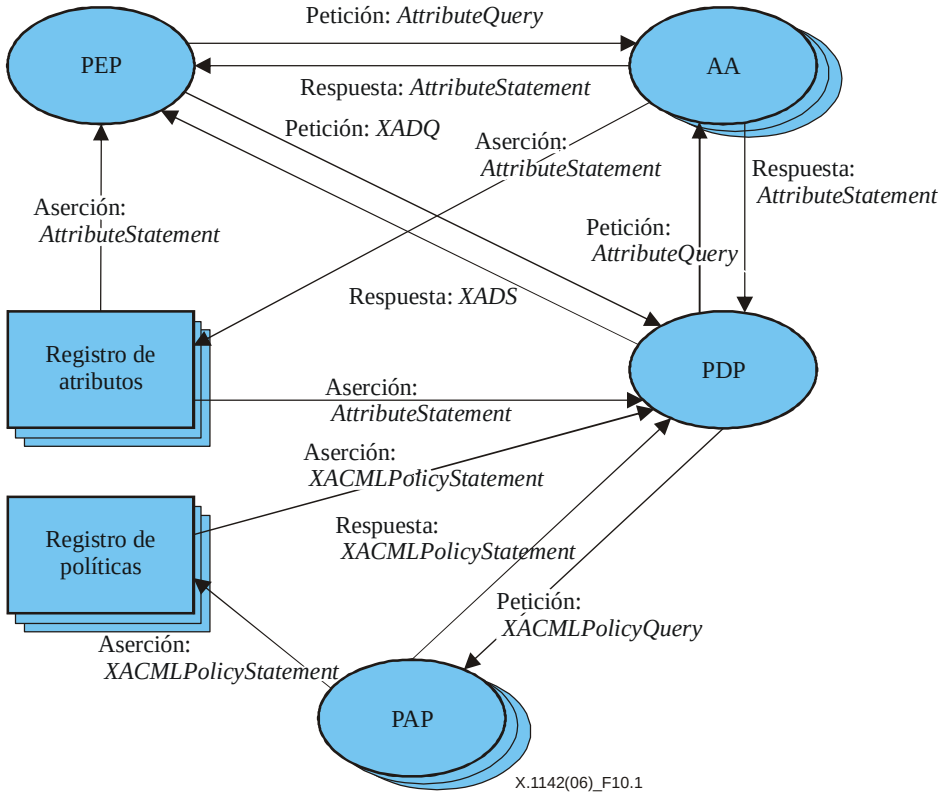

Figura 10-1/X.1142 – Modelo de utilización XACML

En esta cláusula se describen todos los elementos del esquema de peticiones y aserciones, se explica su utilización y otros aspectos del uso de SAML con XACML. Esta Recomendación no requiere cambios o extensiones a XACML, si bien define extensiones a SAML.

Para mejorar la legibilidad, los ejemplos de este perfil están basados en la hipótesis de utilización de las siguientes declaraciones de entidades internas XML:

```
^lt;!ENTITY saml "urn:oasis:names:tc:SAML:2.0:assertion"
^lt;!ENTITY samlp "urn:oasis:names:tc:SAML:2.0:protocol"
^lt;!ENTITY xacml "urn:oasis:names:tc:xacml:2.0:"
```

```
^lt;!ENTITY xacml-context "urn:oasis:names:tc:xacml:2.0:context:schema:os"
^lt;!ENTITY xml "http://www.w3.org/2001/XMLSchema#"
^lt;!ENTITY subject-id "urn:oasis:names:tc:xacml:1.0:subject:subject-id"
^lt;!ENTITY resource "urn:oasis:names:tc:xacml:1.0:resource:"
^lt;!ENTITY resource-id "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
^lt;!ENTITY action-id "urn:oasis:names:tc:xacml:1.0:action:action-id"
^lt;!ENTITY environment "urn:oasis:names:tc:xacml:1.0:environment:"
^lt;!ENTITY current-dateTime
    "urn:oasis:names:tc:xacml:1.0:environment:current-dateTime"
```

Por ejemplo, "`&xml;#string`" es equivalente a http://www.w3.org/2001/XMLSchema#string. El espacio de nombres correspondiente al esquema XACML de extensión de un esquema de aserción SAML es:

```
xacml-saml="urn:oasis:names:tc:xacml:2.0:saml:assertion:schema:os"
```

El espacio de nombres correspondiente al esquema XACML de extensión de un esquema de protocolo SAML es:

```
xacml-samlp="urn:oasis:names:tc:xacml:2.0:saml:protocol:schema:os"
```

10.1 Correspondencia entre atributos SAML y XACML

El esquema de aserciones SAML define una aserción de atributos. El esquema de protocolo SAML define una petición `AttributeQuery` que se utiliza para solicitar ejemplares de aserciones de atributos, y una respuesta que contiene los ejemplares solicitados. Los sistemas que usan XACML pueden usar ejemplares de estos elementos SAML para transmitir y almacenar atributos SAML, y también pueden usar el protocolo `AttributeQuery` SAML para solicitar ejemplares de atributos SAML. Para poder utilizar un atributo SAML en el contexto Petición XACML es necesario transcribirlo en un atributo XACML.

Una aserción de atributo SAML es un ejemplar `<saml:Assertion>` que contiene uno o varios ejemplares `<saml:AttributeStatement>`, cada una de las cuales puede contener uno o varios ejemplares `<saml:Attribute>`.

Para poder utilizarlo en el contexto Petición XACML, cada atributo SAML en la aserción de atributos SAML deberá ajustarse al *Perfil de atributos XACML*, que tiene el siguiente espacio de nombre: `urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML`, en la Rec. UIT-T X.1141.

Los elementos `<xacml-context:Attribute>` se reformarán a partir del elemento `<saml:Attribute>` correspondiente de una aserción de atributos SAML como se indica a continuación:

- Atributo XML de identificador (`AttributeId`) para XACML
  - Se debe usar el valor totalmente cualificado del atributo XML de nombre `<saml:Attribute>`.
- Atributo XML de `DataType` para XACML
  - Se debe usar el valor totalmente cualificado del atributo XML de `DataType` `<saml:Attribute>`. Si no se ha incluido, el atributo XML de `DataType` `<saml:Attribute>` para XACML será: http://www.w3.org/2001/XMLSchema#string.
- Atributo XML de emisor para XACML
  - Se debe usar el valor de cadena del elemento `<saml:Issuer>` que aparece en la Aserción de Atributo SAML.
- `<xacml-context:AttributeValue>`
  - Se debe usar el valor `<saml:AttributeValue>` para el elemento `<xacml-context:AttributeValue>`.

Cada ejemplar `<saml:Attribute>` se transcribe en un solo elemento `<xacml-context:Attribute>`. No es necesario transcribir todos los ejemplares `<saml:Attribute>` de una aserción de atributo SAML; los ejemplares de atributo SAML que es necesario transcribir se pueden seleccionar mediante un mecanismo no especificado aquí. El `Issuer` de un elemento `<saml:Assertion>` será el `Issuer` de cada elemento `<xacml-context:Attribute>` que se crea.

El `<xacml-context:Attribute>` creado a partir de `<saml:Assertion>` se debe situar en el elemento `<xacml-context:Resource>`, `<xacml-context:Subject>`, `<xacml-context:Action>`, o `<xacml-context:Environment>` que corresponde a una entidad que es el `<saml:Subject>` en la aserción de atributos SAML. Por ejemplo, si el sujeto de la aserción de atributos SAML contiene un elemento `<saml:NameIdentifier>`, y el valor de ese `NameIdentifier` coincide con el valor de

<xacml-context:Attribute> que tiene un identificador (AttributeId) &resource;resource-id, entonces los ejemplares <xacml-context:Attribute> creadas a partir de <saml:Attribute> de la aserción de atributos SAML se utilizarán en el elemento <xacml-context:Resource>. Si <xacml context:Attribute> se utiliza en un elemento <xacml-context:Subject>, el atributo XML de categoría (SubjectCategory) para XACML también debe ser coherente con la entidad que es el sujeto de la <saml:Assertion>.

La entidad que realiza la transcripción garantizará que se respeta la semántica definida por SAML para los elementos en la <saml:Assertion>. La entidad transcriptora no tiene que comprobar por sí misma la semántica, pero debe asegurar que las comprobaciones se han realizado antes de que un PDP de XACML utilice un <xacml:Attribute> creado a partir de la <saml:Assertion>. Estas comprobaciones de semántica incluyen, sin ser exhaustivas, las siguientes:

- Cualquier atributo XML NotBefore y NotOnOrAfter en la <saml:Assertion> debe ser válido con respecto a la <xacml:Request> en la que se usa el <xacml:Attribute> derivado de SAML. Esto significa que los valores de los atributos XML NotBefore y NotOnOrAfter deben ser coherentes con los valores del <xacml:Attribute> &environment;current-time, &environment;current-date, y &environment:current-dateTime definidos para la petición <xacml:Request>.
- La entidad transcriptora garantizará que se respeta la semántica definida por SAML para cualquier elemento <saml:AudienceRestrictionCondition> o <saml:DoNotCacheCondition>.
- Si hay un elemento <ds:Signature> en la <saml:Assertion>, entonces la entidad transcriptora garantizará que la firma es válida y que el elemento SAML <Issuer> es coherente con cualquier valor <ds:X509IssuerName> en la firma. Hay que observar las directrices referentes a las firmas digitales de la Rec. UIT-T X.1141.

10.2 Decisiones de autorización

SAML 2.0 define una petición de decisión de autorización rudimentaria (AuthzDecisionQuery) (véase la Rec. UIT-T X.1141). Una AuthzDecisionQuery SAML es incapaz de transmitir toda la información que un PDP XACML puede aceptar como parte de su contexto de petición. Asimismo, la declaración de decisión de autorización (AuthzDecisionStatement) SAML es incapaz de transmitir toda la información contenida en un contexto de respuesta XACML.

Para permitir a un PEP usar la sintaxis de petición y respuesta SAML con pleno soporte del contexto Petición XACML y la sintaxis del contexto de respuesta, en esta Recomendación se definen dos extensiones SAML:

- <xacml-samlp:XACMLAuthzDecisionQuery> es una petición SAML que extiende el esquema de protocolo SAML (véase la Rec. UIT-T X.1141). Permite que el PEP presente un contexto de Petición XACML en una petición SAML, junto con otra información.
- <xacml-saml:XACMLAuthzDecisionStatement> es una declaración SAML que extiende el esquema de aserción SAML (véase la Rec. UIT-T X.1141). Permite que un PDP XACML devuelva un contexto de respuesta XACML en la respuesta a una <XACMLAuthzDecisionStatement>, junto con otra información. También permite que se almacene o transmita un contexto de respuesta XACML en forma de aserción SAML.

10.2.1 Elemento <XACMLAuthzDecisionQuery>

El elemento <XACMLAuthzDecisionQuery> puede ser usado por un PEP para solicitar una decisión de autorización de un PDP XACML. Permite transmitir un ejemplar de contexto de Petición XACML en una petición SAML.

```
<xs:element name="XACMLAuthzDecisionQuery" type="XACMLAuthzDecisionQueryType"/>
<xs:complexType name="XACMLAuthzDecisionQueryType">
   <xs:complexContent>
         <xs:extension base="samlp:RequestAbstractType">
               <xs:sequence>
                     <xs:element ref="xacml-context:Request"/>
               </xs:sequence>
               <xs:attribute name="InputContextOnly" type="boolean"
use="optional" default="false"/>
               <xs:attribute name="ReturnContext" type="boolean" use="optional"
default="false"/>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <XACMLAuthzDecisionQuery> es del tipo complejo **XACMLAuthzDecisionQueryType**. Este elemento es una alternativa a la petición <samlp:AuthzDecisionQuery> definida en SAML, que permite a un PEP utilizar todas las capacidades de un PDP XACML.

El elemento <XACMLAuthzDecisionQuery> contiene los siguientes elementos y atributos XML:

– InputContextOnly [Por defecto "false"]

  Este atributo XML determina las fuentes de información que el PDP podrá utilizar al tomar su decisión de autorización. Si el valor de dicho atributo XML es "true", entonces la decisión de autorización se debe tomar solamente basándose en la información contenida en <XACMLAuthzDecisionQuery>; no se pueden utilizar atributos externos. Si el valor de dicho atributo XML es "false", entonces la decisión de autorización puede basarse en atributos externos no contenidos en la <XACMLAuthzDecisionQuery>.

– ReturnContext [Por defecto "false"]

  Este atributo XML permite al PEP solicitar que se incluya un elemento <xacml-context:Request> en la <XACMLAuthzDecisionStatement> resultante de la petición. También determina el contenido del elemento <xacml-context:Request>.

  Si el valor de este atributo XML es "true", el PDP debe incluir el elemento <xacml-context:Request> en el elemento <XACMLAuthzDecisionStatement> en la respuesta <XACMLResponse>. Este elemento <xacml-context:Request> debe incluir todos los atributos aportados por el PEP en la petición <XACMLAuthzDecisionQuery> que se tomaron en cuenta para la decisión de autorización. El PDP puede incluir atributos adicionales en este elemento <xacml-context:Request>, tales como atributos externos obtenidos por el PDP y que se tomaron en cuenta para la decisión de autorización, u otros atributos conocidos por el PDP, que pueden ser útiles para el PEP en ulteriores peticiones <XACMLAuthzDecisionQuery>.

  Si el valor de este atributo XML es "false", el PDP no debe incluir el elemento <xacml-context:Request> en el elemento <XACMLAuthzDecisionStatement> de la respuesta <XACMLResponse>.

– <xacml-context:Request> [Necesario]

  Un contexto de Petición XACML.

10.2.2 Elemento <XACMLAuthzDecisionStatement>

Un PDP en XACML puede utilizar la declaración <XACMLAuthzDecisionStatement> para devolver una respuesta SAML que contenga un contexto de respuesta XACML a un PEP, tras una petición <XACMLAuthzDecisionQuery>. También se puede usar en una aserción SAML como formato para el almacenamiento de una decisión de autorización en un registro.

```
<xs:element name="XACMLAuthzDecisionStatement" type="xacml-
saml:XACMLAuthzDecisionStatementType"/>
<xs:complexType name="XACMLAuthzDecisionStatementType">
   <xs:complexContent>
         <xs:extension base="saml:StatementAbstractType">
               <xs:sequence>
                     <xs:element ref="xacml-context:Response"/>
                           <xs:element ref="xacml-context:Request"
MinOccurs="0"/>
               </xs:sequence>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <XACMLAuthzDecisionStatement> es del tipo complejo **XACMLAuthzDecisionStatementType**. Este elemento es una alternativa a la declaración <samlp:AuthzDecisionStatement> definida en SAML, que permite incluir en una aserción SAML el contenido completo de la respuesta de un PDP XACML.

El elemento <XACMLAuthzDecisionStatement> contiene los elementos siguientes:

– <xacml-context:Response> [Necesario]

  El contexto de respuesta XACML creado por el PDP XACML en respuesta a una petición <XACMLAuthzDecisionQuery>.

– <xacml-context:Request> [Facultativo]

  Un elemento <xacml-context:Request> que contiene atributos XACML devueltos por el PDP XACML en respuesta a la petición <XACMLAuthzDecisionQuery>. Habrá que incluir este elemento si el valor del

atributo XML `ReturnResponse` en `<XACMLAuthzDecisionQuery>` es "true". No se incluirá si el atributo XML `ReturnResponse` en `<XACMLAuthzDecisionQuery>` es "false".

10.3 Políticas

XACML define dos elementos del esquema de políticas: `<Policy>` y `<PolicySet>`. SAML no define protocolos ni esquemas de aserción para políticas. Esta cláusula define nuevas extensiones SAML para los elementos `<XACMLPolicyQuery>` y `<XACMLPolicyStatement>`. Se pueden usar ejemplares de estos nuevos elementos para solicitar, transmitir y almacenar ejemplares XACML `<Policy>` y `<PolicySet>`.

10.3.1 Elemento <XACMLPolicyQuery>

El elemento `<XACMLPolicyQuery>` lo utiliza un PDP para solicitar uno o varios ejemplares XACML de política (Policy) o conjunto de políticas (`PolicySet`) a partir de un punto de administración de políticas en línea, como parte de la petición SAML.

```
<xs:element name="XACMLPolicyQuery" type="XACMLPolicyQueryType"/>
<xs:complexType name="XACMLPolicyQueryType">
   <complexContent>
         <xs:extension base="samlp:RequestAbstractType">
               <xs:choice minOccurs="0" maxOccurs="unbounded">
                     <xs:element ref="xacml-context:Request"/>
                     <xs:element ref="xacml:Target"/>
                     <xs:element ref="xacml:PolicySetIdReference"/>
                     <xs:element ref="xacml:PolicyIdReference"/>
               </xs:choice>
         </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento `<XACMLPolicyQuery>` es del tipo complejo **XACMLPolicyQueryType**.

El elemento `<XACMLPolicyQuery>` contiene uno o varios de los siguientes elementos:

– `<xacml-context:Request>` [Cualquier número]

  Suministra un contexto de Petición XACML. Hay que devolver todos los ejemplares XACML policy y `PolicySet` aplicables a esta petición.

– `<xacml:Target>` [Cualquier número]

  Suministra un elemento `<Target>` XACML. Hay que devolver todos los ejemplares XACML policy y `PolicySet` aplicables a este `<Target>`.

– `<xacml:PolicySetIdReference>` [Cualquier número]

  Identifica un `<PolicySet>` XACML que es necesario devolver.

– `<xacml:PolicyIdReference>` [Cualquier número]

  Identifica un `<Policy>` XACML que es necesario devolver.

10.3.2 Elemento <XACMLPolicyStatement>

El elemento `<XACMLPolicyStatement>` lo utiliza un punto de administración de políticas para devolver uno o varios ejemplares XACML `<Policy>` o `<PolicySet>` en una respuesta SAML a una petición SAML `<XACMLPolicyQuery>`. El elemento `<XACMLPolicyStatement>` también se puede usar en una aserción SAML como formato para almacenar `<XACMLPolicyStatement>` en un registro.

```
<xs:element name="XACMLPolicyStatement" type="xacml-
saml:XACMLPolicyStatementType"/>
<xs:complexType name="XACMLPolicyStatementType">
   <xs:complexContent>
         <xs:extension base="saml:StatementAbstractType">
               <xs:choice minOccurs="0" maxOccurs=unbounded">
                     <xs:element ref="xacml:Policy"/>
                     <xs:element ref="xacmlPolicySet"/>
               </xs:choice>
               </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

El elemento <XACMLPolicyStatement> es del tipo complejo **XACMLPolicyStatementType**.

El elemento <XACMLPolicyStatement> contiene los siguientes elementos. Si <XACMLPolicyStatement> se produce como respuesta a una petición <XACMLPolicyQuery>, y no hay ejemplares <xacml:Policy> o <xacml:PolicySet> que cumplan las especificaciones de la <XACMLPolicyQuery> asociada, no habrá ningún elemento en <XACMLPolicyStatement>.

– <xacml:Policy> [Cualquier número]

  Un ejemplar <xacml:Policy> que cumple las especificaciones de la <XACMLPolicyQuery>, en su caso.

– <xacml:PolicySet> [Cualquier número]

  Un ejemplar <xacml:PolicySet> que cumple las especificaciones de la <XACMLPolicyQuery>, en su caso.

10.4 Elemento <saml:Assertion>

Las declaraciones <XACMLAuthzDecisionStatement>, <XACMLPolicyStatement>, o <saml:AttributeStatement> SAML normal se deben encapsular en una <saml:Assertion>, que puede estar firmada.

La mayoría de componentes de una <saml:Assertion> están completamente especificados en la Rec UIT-T X.1141. Los siguientes elementos y atributos XML se especifican aquí además para el uso con los tipos de declaraciones SAML definidos y usados en este perfil.

Excepto lo especificado aquí, este perfil no impone requerimientos o restricciones de información en el elemento <saml:Assertion>.

10.4.1 Elemento <saml:Issuer>

El elemento <saml:Issuer> obligatorio informa sobre "la autoridad SAML que afirma algo en la aserción".

Para soportar las firmas digitales de terceros, este perfil no requiere que la identidad proporcionada en el elemento <saml:Issuer> sea coherente con la identidad del firmante. La parte confiante debe tener una relación de confianza apropiada con la autoridad que firma la <saml:Assertion>.

Cuando se usa una <saml:AttributeAssertion> para construir un atributo XACML, el valor de la cadena del elemento <saml:Issuer> se usará como valor del atributo XML del emisor en XACML. Es conveniente tenerlo en cuenta al especificar el valor SAML.

10.4.2 Elemento <ds:Signature>

El elemento <ds:Signature> opcional contiene "una firma XML que autentica la aserción".

Se puede usar un elemento <ds:Signature> en una aserción con una declaración XACML. Para soportar las firmas digitales de terceros, este perfil no requiere que la identidad proporcionada en el elemento <saml:Issuer> sea coherente con la identidad del firmante. La parte confiante debe tener una relación de confianza apropiada con la autoridad que firma la <saml:Assertion>.

La parte que confía debería verificar cualquier firma incluida en la aserción y no debería usar información obtenida de la aserción a menos que la firma sea comprobada satisfactoriamente.

10.4.3 Elemento <saml:Subject>

El elemento <saml:Subject> opcional contiene "el sujeto de la(s) declaración(es) en la aserción".

El elemento <saml:Subject> no se debe incluir en una aserción que contenga <XACMLAuthzDecision> o <XACMLPolicy>.

En una aserción <saml:AttributeAssertion> que se va a transcribir en un atributo XACML**,** el elemento <saml:Subject> debe contener la identidad de la entidad a la que corresponden el atributo y su valor. Para un atributo <Subject> en XACML, esta identidad debe ser coherente con el valor de cualquier Atributo &subject-id; en XACML presente en el mismo elemento <Subject>.

Para un atributo <Resource> en XACML, esta identidad debe ser coherente con el valor de cualquier atributo &resource-id; en XACML presente en el mismo elemento <Resource>. Para un atributo <Action> en XACML**,** esta identidad debe ser coherente con el valor de cualquier atributo &action-id; en XACML presente en el mismo

elemento <Action>. Para un atributo <Environment> en XACML, esta identidad debe ser coherente con el valor de cualquier atributo XACML presente en el mismo elemento <Environment> y proporciona una identidad de entorno.

10.4.4 Elemento <saml:Conditions>

El elemento <saml:Conditions> opcional contiene "condiciones que se deben tener en cuenta al valorar la validez de la aserción y/o utilizarla".

El elemento <saml:Conditions> debe contener atributos NotBefore y NotOnOrAfter XML para especificar los límites en la validez de la aserción. Si están presentes estos atributos XML la parte que confía debería asegurarse de que la información que obtiene de la aserción es utilizada por un PDP para evaluar políticas sólo cuando el valor del atributo de recurso &current-dateTime; del contexto de petición está contenido dentro del periodo de validez especificado de la aserción.

10.5 Elemento <samlp:RequestAbstractType>

Las peticiones <XACMLAuthzDecisionQuery> o <XACMLPolicyQuery> deben encapsularse en un elemento <samlp:RequestAbstractType>, el cual puede estar firmado.

La mayoría de componentes de <samlp:RequestAbstractType> están completamente especificados en la Rec. UIT-T X.1141. Para utilizarlo con los tipos de peticiones SAML definidos y usados en este perfil, es necesario observar las condiciones de utilización de los elementos <saml:Issuer> y <ds:Signature> especificadas en la cláusula anterior. Excepto lo especificado aquí, este perfil no impone requerimientos ni restricciones de información en el elemento <samlp:RequestAbstractType>.

10.5.1 Elemento <saml:Issuer>

Véase 10.4.1, Elemento <saml:Issuer>.

10.5.2 Elemento <ds:Signature>

Véase 10.4.2, Elemento <ds:Signature>.

10.6 Elemento <samlp:Response>

Los elementos <XACMLAuthzDecisionStatement> o <XACMLPolicyStatement> deben encapsularse en un elemento <samlp:Response>, el cual puede estar firmado.

La mayoría de componentes de <samlp:Response> están completamente especificados en la Rec. UIT-T X.1141. Los siguientes elementos y atributos XML se especifican aquí además para usarlos con los tipos de declaración SAML definidos y usados en este perfil. Excepto lo especificado aquí, este perfil no impone requerimientos ni restricciones de información en el elemento <samlp:Response>.

10.6.1 Elemento <samlp:Issuer>

Véase 10.4.1, Elemento <saml:Issuer>.

10.6.2 Elemento <ds:Signature>

Véase 10.4.2, Elemento <ds:Signature>.

10.6.3 Elemento <samlp:StatusCode>

El elemento <samlp:StatusCode> es un componente del elemento <samlp:Status> en la <samlp:Response>.

10.6.3.1 Respuesta a una petición <XACMLAuthzDecisionQuery>

En la respuesta a una petición <XACMLAuthzDecisionQuery>, el atributo XML que indica el valor de <samlp:StatusCode> dependerá del elemento <xacml:StatusCode> del elemento <xacml:Status> de la decisión de autorización, según las siguientes reglas:

1) urn:oasis:names:tc:SAML:2.0:status:Success

   Se utilizará este atributo de valor XML para <samlp:StatusCode> sólo si el valor de <xacml:StatusCode> es urn:oasis:names:tc:xacml:1.0:status:ok.

2) `urn:oasis:names:tc:SAML:2.0:status:Requester`

Se utilizará este atributo de valor XML para `<samlp:StatusCode>` cuando el valor de `<xacml:StatusCode>` sea `urn:oasis:names:tc:xacml:1.0:status:missing-attribute` o el valor de `<xacml:StatusCode>` sea `urn:oasis:names:tc:xacml:1.0:status:syntax-error`, debido a un error de sintaxis en la petición `<xacml:Request>`.

3) `urn:oasis:names:tc:SAML:2.0:status:Responder`

Se utilizará este atributo de valor XML para `<samlp:StatusCode>` cuando el valor de `<xacml:StatusCode>` sea `urn:oasis:names:tc:xacml:1.0:status:syntax-error`, debido a un error de sintaxis en una `<xacml:Policy>` o `<xacml:PolicySet>`. Hay que tener en cuenta que no se detectarán todos los errores de sintaxis en políticas al procesar una petición, de manera que no todos los errores de sintaxis de políticas se comunicarán de esta manera.

4) `urn:oasis:names:tc:SAML:2.0:status:VersionMismatch`

Se utilizará este atributo de valor XML para `<samlp:StatusCode>` sólo cuando la interfaz SAML en el PDP no soporte la versión del mensaje de petición SAML utilizado en la búsqueda.

10.6.3.2 Respuesta a <XACMLPolicyQuery>

En la respuesta a una petición de `<XACMLPolicyQuery>`, el atributo de valor XML para `<samlp:StatusCode>` será conforme a las especificaciones de la Rec. UIT-T X.1141.

11 Perfil de firma digital XML

Esta cláusula proporciona un perfil para el uso con la firma digital Signature:2002 del W3C para la autenticación y la protección de integridad de instancias en el esquema XACML.

La firma digital es útil para la autenticación y protección de integridad sólo si la información firmada incluye una especificación de la identidad del firmante y una especificación del periodo de validez del objeto de datos firmado. En el lenguaje XACML no se ha definido el formato para esta información porque está previsto utilizar otras normas en XACML para funciones distintas de la especificación y evaluación de políticas, peticiones y respuestas de control de acceso.

Se ha definido un formato apropiado en SAML. En la cláusula 10 se define un perfil para el uso de SAML con ejemplares del esquema XACML. Por lo tanto, este perfil recomienda el uso de ejemplares del esquema XACML en Aserciones, Peticiones y Respuestas SAML, las cuales se pueden firmar digitalmente como se especifica en la Rec. UIT-T X.1141.

11.1 Uso de SAML

Este perfil recomienda el uso de ejemplares del esquema XACML insertadas en Aserciones, Peticiones y Respuestas SAML, como se describe en la cláusula 10. Tales objetos SAML deben estar firmados digitalmente como se describe en 8.4/X.1141, *Sintaxis y procesamiento de firmas SAML y XML*.

11.2 Canonicalización

A fin de que una firma digital sea verificada por la parte que confía, la cadena de bytes que se firmó debe ser idéntica a la cadena de bytes que se verifica. La solución es canonicalizar el documento XML que se firma (véase Canonicalization:2002 del W3C). En la Rec. UIT-T X.1141 se especifica una canonicalización exclusiva (véase Canonicalization:2002 del W3C).

11.2.1 Elementos de espacio de nombre (Namespace) en objetos de datos XACML

Cualquier objeto de datos XACML que se va a firmar debe especificar todos los elementos namespace usados en el objeto de datos. Si esto no se hace, el objeto de datos atraerá definiciones namespace de progenitores del objeto de datos que pueden diferir de un sobre a otro.

Cuando se utiliza la canonicalización exclusiva como método de regularización, el namespace de esquemas XACML usados por elementos en un objeto de datos XACML se asociará a un prefijo y se incluirá en el parámetro `InclusiveNamespacesPrefixList` para http://www.w3.org/2001/10/xml-exc-c14n# (véase Canonicalization:2002 del W3C).

11.2.2 Consideraciones adicionales sobre la canonicalización

Normalmente se deben realizar transformaciones adicionales en el objeto de datos XACML para asegurar que el objeto de datos firmado coincidirá con el objeto de datos verificado. Algunas de estas transformaciones se indican aquí en una lista, pero en este perfil no se especificarán algoritmos para realizarlas.

Si un objeto de datos XACML incluye elementos de datos que se pueden representar de más de una forma (por ejemplo (TRUE, FALSE), (1,0), (true,false)), entonces se tiene que definir y especificar un método de transformación para normalizar estos elementos de datos.

Este perfil recomienda aplicar las canonicalizaciones siguientes a los valores de los correspondientes tipos de datos, siendo indiferente que aparezcan en valores de atributo XML o en atributos XACML.

1) Si se define una representación canónica para un tipo de datos definido en XACML en http://www.w3.org/2001/XMLSchema, habrá que poner el valor del tipo de datos en la forma canónica especificada en http://www.w3.org/2001/XMLSchema. Esto incluye booleano {"true", "false"}, doble, dateTime, tiempo, fecha, y hexBinary (mayúscula).
2) http://www.w3.org/2001/XMLSchema#anyURI – Debe utilizarse la forma canónica definida en IETF RFC 2396.
3) http://www.w3.org/2001/XMLSchema#base64Binary – Suprimir todos los saltos de línea y espacios en blanco. Suprimir todos los caracteres que siguen a la primera secuencia de "=" caracteres. La transformación Base64 (identificador: http://www.w3.org/TR/xmldsig-core/#sec-Base-64) puede ser útil para efectuar esta canonicalización.
4) `urn:oasis:names:tc:xacml:1.0:data-type:x500Name` – Normalizar primero según IETF RFC 2253. Si cualquier RDN contiene múltiples pares `attributeTypeAndValue`, reordenar los `AttributeValuePairs` en esa RDN en orden ascendente, comparados como cadenas de octetos (véase 11.6/X.690).
5) `urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name` – Normalizar la parte del dominio del nombre a minúsculas.
6) expresión XPath – Aplicar http://www.w3.org/2002/06/xmldsig-filter2 para poner la expresión XPath en forma canónica.

11.3 Esquemas de firmas

El análisis de cualquier objeto de datos XACML necesita una copia exacta de todos los esquemas de los que depende el objeto de datos XACML. Hay que tener en cuenta que la inclusión de un URI de esquema en los atributos del ejemplar del esquema XACML no garantiza que se va a utilizar una copia exacta del esquema: un atacante puede sustituir un esquema ficticio que contenga el identificador correcto. Las firmas pueden ayudar a proteger contra la sustitución o modificación de los esquemas de los que depende el objeto de datos XACML. En esta cláusula se describe el uso de firmas con este propósito.

En la mayoría de casos, un firmante del objeto de datos debe incluir un elemento `<Reference>` para cada esquema del que depende el objeto de datos XACML, en el elemento `<SignedInfo>` que contiene la `<Reference>` al objeto de datos XACML o que incluye el objeto propiamente dicho.

En algunos casos, el firmante del objeto de datos sabe que todos los PDP que evaluarán un objeto de datos XACML dado tendrán copias exactas de ciertos esquemas necesarios para analizar sintácticamente el objeto de datos, y no quiere obligar al PDP a verificar el compendio de mensajes de tales esquemas. En estos casos el firmante del objeto de datos puede omitir los elementos `<Reference>` para cualquier esquema cuya verificación no sea necesaria.

12 Perfil de recursos jerárquicos de XACML

A menudo se da el caso que un recurso se organiza como una jerarquía: por ejemplo, sistemas de archivos, documentos XML y organizaciones. Esta cláusula especifica cómo XACML puede proporcionar control de acceso para un recurso organizado como una jerarquía.

¿Por qué son especiales los recursos organizados como jerarquías? En primer lugar, las políticas de jerarquías aplican frecuentemente los mismos controles de acceso a subárboles completos de la jerarquía. Si se puede expresar una sola restricción de políticas que se aplicará a un subárbol entero de nodos en la jerarquía, en lugar de tener que especificar una restricción separada para cada nodo, es más fácil de utilizar y más probable que la política reflejará correctamente los controles de acceso deseados. Otra característica especial de los recursos jerárquicos es que el acceso a un nodo puede depender del valor de otro nodo. Por ejemplo, un paciente puede tener acceso al nodo "diagnóstico" en un archivo de documentos médicos, sólo si el nombre del paciente coincide con el valor en el nodo "nombre del paciente".

Si éste es el caso, el nodo solicitado no puede ser procesado de forma aislada del resto de nodos de la jerarquía, y el PDP debe tener acceso a los valores de otros nodos. Finalmente, la identidad de los nodos en una jerarquía depende a menudo de la posición del nodo en la jerarquía. También puede haber múltiples formas de describir la identidad de un solo nodo. Para aplicar las políticas a los nodos, es importante que las representaciones de identidad de los nodos sean coherentes. De otro modo, un solicitante puede saltarse los controles de acceso, solicitando un nodo con una identidad diferente de la utilizada por la política.

Un recurso organizado como una jerarquía puede ser un "árbol" (una jerarquía con una sola raíz) o un "bosque" (una jerarquía con múltiples raíces), pero la jerarquía no puede tener ciclos. Estos dos tipos de jerarquías también se conocen como "gráficos acíclicos dirigidos" (DAG, *directed acyclic graph*) o "DAG". Todos estos recursos son denominados recursos jerárquicos en este perfil. Un documento XML se estructura siempre como un "árbol". Otros tipos de recursos jerárquicos se pueden estructurar como "bosques", por ejemplo los archivos en un sistema de archivos que soporta enlaces.

Los nodos en un recurso jerárquico son tratados como recursos individuales. Una decisión de autorización que permite acceso a un nodo interior no implica que esté permitido el acceso a sus nodos descendientes. Una decisión de autorización que deniega el acceso a un nodo interior no implica que se deniegue el acceso a sus nodos descendientes.

En este perfil se especifican tres procedimientos que se aplican a los recursos jerárquicos:

- Representación de la identidad de un nodo.
- Solicitud de acceso a un nodo.
- Definición de políticas que se aplican a uno o varios nodos.

No es obligatorio que la implementación soporte estos procedimientos.

En esta cláusula se consideran dos formas de representar un recurso jerárquico. De la primera forma, la jerarquía de la que es parte el nodo se representa como un documento XML que se incluye en la petición, y el recurso solicitado se representa como un nodo en ese documento. De la segunda forma, el recurso solicitado no se representa como un nodo en un documento XML y en la petición no hay ninguna representación de la jerarquía de la cual forma parte. Obsérvese que el recurso considerado en el primer caso no necesita formar parte de un documento XML – sólo se representa de esa forma en la Petición. Asimismo, el recurso considerado en el segundo caso puede formar parte de un documento XML, pero se representa de otra forma en la Petición. Por tanto, no se supone que haya una correlación entre la estructura del recurso como se representa en la Petición y la estructura efectiva del recurso físico al que se accede.

Los procedimientos para tratar con recursos representados como nodos en documentos XML pueden hacer uso del hecho de que el propio documento XML está incluido en la petición de decisión. Las expresiones Xpath se pueden usar para normalizar las referencias a nodos en este documento, y pueden proporcionar representaciones únicas para un nodo dado en el documento. No existen procedimientos similares para recursos jerárquicos que no se representan como documentos XML. Es necesario proporcionar otras soluciones en el caso de los recursos no XML, para determinar la localización del nodo solicitado en la jerarquía. En algunos casos se puede incluir la posición del nodo en la jerarquía como parte de su identidad. En otros casos, un nodo puede tener más de una identidad normativa, como cuando el nombre de la ruta de un archivo en un sistema de archivos puede incluir enlaces duros. En tales casos, la función de servicio del contexto del PDP XACML probablemente tenga que proporcionar las identidades de todos los nodos progenitores. Por todas estas razones, los procedimientos para tratar con nodos en documentos XML difieren de los procedimientos para tratar con nodos en otros recursos jerárquicos.

Cuando se trata de un recurso jerárquico, puede ser útil solicitar decisiones de autorización para múltiples nodos en el recurso en una única petición de decisión. Puede considerarse que esta cláusula se superpone al perfil Recurso Múltiple (véase la cláusula 9), que a su vez se superpone al comportamiento especificado en la cláusula 7. Sin embargo, la funcionalidad en esta cláusula puede estar directamente sobre la funcionalidad de la cláusula 7.

En esta cláusula para recursos jerárquicos se supone que todas las peticiones de acceso a nodos múltiples en un recurso jerárquico se han convertido en peticiones individuales de acceso a un nodo único.

12.1 Representación de la identidad de un nodo

Para que la aplicación de las políticas XACML sea coherente en los nodos de un recurso jerárquico, es necesario que los nodos de dicho recurso estén representados de forma coherente. Si una política se refiere a un nodo que usa una representación, pero una petición se refiere al nodo usando una representación diferente, entonces la política no se aplicará y tal vez no se pueda garantizar la seguridad.

Las cláusulas siguientes describen representaciones recomendadas para nodos en recursos jerárquicos. Es posible utilizar representaciones alternativas de nodos en un recurso dado si todos los puntos de administración de políticas y todos los puntos de ejecución de políticas que tratan con ese recurso han acordado utilizar la representación alternativa.

12.1.1 Nodos en documentos XML

Esta cláusula es normativa, pero es facultativa.

El siguiente URI se debe utilizar como identificador para la funcionalidad especificada en esta parte de este perfil:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-id
```

La identidad de un nodo en un recurso representado como un ejemplar de documento XML debe ser una expresión XPath cuyo resultado de evaluación sea exactamente ese nodo en la copia del recurso que se incluye en el elemento `<ResourceContent>` del elemento `<Resource>` de la Petición `<Request>`.

12.1.2 Nodos en recursos que no son documentos XML

Esta cláusula es normativa, pero es facultativa.

El siguiente URI se debe utilizar como identificador para la funcionalidad especificada en esta parte de este perfil:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-id
```

La identidad de un nodo en un recurso jerárquico que no se representa como un ejemplar de documento XML ha de ser un URI conforme a IETF RFC 2396. Estos URI tienen la forma siguiente.

`<scheme>` ":" `<authority>` "/" `<pathname>`

Los recursos de sistema de archivo deben utilizar el esquema "file:". Si el documento IETF RFC 2396 o una norma relacionada para esquemas URI registrados no especifican un `<scheme>` normal para el tipo de recurso, entonces el URI debe usar el esquema "`file`:".

El fragmento `<pathname>` del URI tendrá la siguiente forma:

`<root name>` ["/" `<node name>`]*

La secuencia de valores `<root name>` y `<node name>` debe corresponder a los nombres de componentes jerárquicos de progenitores del nodo representado entre el nodo raíz `<root>` y este nodo representado.

Se utilizarán las siguientes reglas de canonicalización:

- La codificación del URI debe ser UTF8.
- Los fragmentos sensibles a mayúsculas o minúsculas del URI deben estar en minúsculas.
- Se aplicarán las normas de caracteres de escape del documento IETF RFC 2396.
- El fragmento `<authority>` del URI debe estar especificado y será la representación de la autoridad normal para el tipo de recurso dado. Cuando la `<authority>` se pueda especificar mediante un servicio de nombres de dominio (DNS) o con una dirección numérica IPv4 o IPv6, se debe usar el nombre DNS.
- Los componentes del fragmento `<pathname>` del URI se deben especificar utilizando la forma canónica para dichos componentes del camino en la `<authority>`.
- Conforme a IETF RFC 2396, el carácter separador entre componentes jerárquicos del fragmento `<pathname>` del URI debe ser el carácter "/". Las secuencias del carácter "/" se deben reducir a un solo "/". Las identidades de nodos no podrán terminar con el carácter "/".
- El `<pathname>` no debe contener enlaces simbólicos.
- Todos los valores del `<pathname>` deben ser absolutos.
- Si hay más de un camino absoluto completamente determinado desde una `<root>` en la `<authority>` hasta el nodo representado, habrá que incluir un atributo de recurso separado que tenga como identificador (`AttributeId`) "`urn:oasis:names:tc:xacml:1.0:resource:resource-id`" y como `DataType http://urn:oasis:names:tc:xacml:1.0:data-type:anyURI` en el Contexto de Petición para cada uno de estos caminos.

12.2 Solicitud de acceso a un nodo

Para que la aplicación de las políticas XACML sea coherente en los nodos de un recurso jerárquico, es necesario que en cada contexto de petición de acceso a un nodo en ese recurso se use una descripción coherente de ese acceso al nodo. Si una política se refiere a ciertos atributos esperados de un nodo, pero el contexto de petición no contiene dichos atributos, o si los atributos no se expresan de la forma esperada, es posible que no se pueda aplicar la política y tal vez no se pueda garantizar la seguridad.

Las cláusulas siguientes describen descripciones recomendadas de contextos de petición de acceso a nodos en recursos jerárquicos. Es posible utilizar representaciones alternativas de dichas peticiones si todos los puntos de administración de políticas y todos los puntos de ejecución de políticas que tratan con ese recurso han acordado utilizar la representación alternativa.

12.2.1 Nodos en documentos XML

Esta cláusula es normativa, pero es opcional.

El siguiente URI se debe utilizar como identificador para la funcionalidad especificada en esta parte de este perfil:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req
```

Los atributos con los identificadores (AttributeId)

```
"urn:oasis::names:tc:xacml:2.0:resource:resource-parent"
"urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor"
```

y:

```
"urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self"
```

son facultativos. Si se implementan en recursos representados como documentos XML, los siguientes URI se deben usar como identificadores para la funcionalidad que representan:

```
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-parent"
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor"
```

y:

```
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-
ancestor-or-self"
```

Para solicitar acceso a un recurso representado como un nodo en un documento XML, el elemento <Resource> del contexto de petición debe contener los siguientes elementos y atributos XML.

- Un elemento <ResourceContent> que contenga el ejemplar completo del documento XML al que pertenece el nodo solicitado.
- Un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:1.0:resource:resource-id" y DataType "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". El valor <AttributeValue> de este <Attribute> debe ser una expresión XPath que tenga como nodo de contexto el único vástago del elemento <ResourceContent>. El resultado de evaluación de esta expresión XPath debe ser un grupo de nodos que contenga el nodo único en el elemento <ResourceContent> que es el nodo al cual se solicita acceso. Este <Attribute> puede especificar un Issuer.
- Un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:2.0:resource:resource-parent" y DataType "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". El <AttributeValue> de este <Attribute> debe ser una expresión XPath; el nodo de contexto debe ser el único vástago del elemento <ResourceContent>. El resultado de evaluación de esta expresión XPath debe ser un grupo de nodos que contenga el nodo único en el elemento <ResourceContent> que es el progenitor inmediato del nodo representado en el atributo "resource-id". Este <Attribute> puede especificar un Issuer.
- Por cada nodo en el ejemplar de documento XML que sea un antecesor del nodo representado por el atributo "resource-id", un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor" y DataType "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". El <AttributeValue> de este <Attribute> debe ser una expresión XPath; el nodo de contexto debe ser el único vástago del elemento <ResourceContent>. El resultado de evaluación de esta expresión XPath debe ser un grupo de nodos que contenga el nodo único en el elemento <ResourceContent> que es el antecesor respectivo del nodo representado en el atributo "resource-id". Para cada atributo "resource-parent" habrá un atributo "resource-ancestor" correspondiente. Este <Attribute> puede especificar un Issuer.

– Por cada nodo en la instancia de documento XML que sea un antecesor del nodo representado por el atributo "resource-id" y también para el propio nodo "resource-id", un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self" y DataType "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". El <AttributeValue> de este <Attribute> debe ser una expresión XPath; el nodo de contexto debe ser el único vástago del elemento <ResourceContent>. El resultado de evaluación de esta expresión XPath debe ser un grupo de nodos que contenga el nodo único en el elemento <ResourceContent> que es el antecesor respectivo del nodo representado en el atributo "resource-id", o el propio nodo "resource-id". Para cada atributo "resource-parent" y "resource-id" habrá un atributo "resource-ancestor-or-self" correspondiente. Este <Attribute> puede especificar un Issuer.

Se pueden incluir atributos adicionales en el elemento <Resource>, en particular el siguiente:

– Un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:2.0:resource:document-id" y DataType "urn:oasis:names:tc:xacml:1.0:data-type:anyURI". El <AttributeValue> de este <Attribute> debe ser un URI que identifique el documento XML al que pertenece el recurso solicitado y que se ha copiado en el elemento <ResourceContent>. Este <Attribute> puede especificar un Issuer.

12.2.2 Nodos en recursos que no son documentos XML

Esta cláusula es normativa, pero es facultativa.

El siguiente URI se debe usar como identificador para la funcionalidad especificada en esta parte de esta cláusula:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req
```

Los atributos con los identificadores (AttributeId):

```
"urn:oasis::names:tc:xacml:2.0:resource:resource-parent"
"urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor"
```

y:

```
"urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self"
```

son facultativos. Si se implementan en recursos que no se representan como documentos XML, los siguientes URI se deben usar como identificadores para la funcionalidad que representan:

```
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-parent"
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor"
```

y:

```
"urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:
resource-ancestor-or-self"
```

Para solicitar acceso a un nodo en un recurso jerárquico que no está representado como un documento XML, el elemento <Resource> del contexto de petición no debe contener un elemento <ResourceContent>. El elemento <Resource> del contexto de petición debe contener los siguientes elementos y atributos XML. Téngase presente que un nodo en un recurso jerárquico que no se representa como un documento XML puede tener múltiples progenitores. Por ejemplo, en un sistema de archivos que admite enlaces duros puede haber múltiples rutas normativas hacia un único archivo. Cada una de dichas rutas puede contener diferentes grupos de progenitores y antecesores.

– Por cada representación normativa del nodo solicitado, un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:1.0:resource:resource-id". El <AttributeValue> de este <Attribute> debe ser la identidad normativa única del nodo al cual se solicita acceso. El DataType de este <Attribute> dependerá de la representación elegida para la identidad de los nodos en este recurso particular. Este <Attribute> puede especificar un Issuer.

– Por cada progenitor inmediato del nodo especificado en el atributo o los atributos "resource-id", y para cada representación normativa de ese nodo progenitor, un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:2.0:resource:resource-parent". El <AttributeValue> de este <Attribute> debe ser la identidad normativa única del nodo progenitor. El DataType de este <Attribute> dependerá de la representación elegida para la identidad de los nodos en este recurso particular. Este <Attribute> puede especificar un Issuer. Si el nodo solicitado es parte de un bosque y no de un solo árbol, o si el nodo progenitor tiene más de una representación normativa, debe haber al menos un ejemplar de este atributo para cada progenitor en el

camino hacia las múltiples raíces de las que desciende el nodo solicitado, y para cada representación de estos progenitores.

– Por cada antecesor del nodo especificado en el atributo o los atributos "resource-id", y para cada representación normativa de ese nodo antecesor, un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor". El <AttributeValue> de este <Attribute> debe ser la identidad normativa única del nodo antecesor. El DataType de este <Attribute> dependerá de la representación elegida para la identidad de los nodos en este recurso particular. Este <Attribute> puede especificar un Issuer. Para cada atributo "resource-parent" habrá un atributo "resource-ancestor" correspondiente. Si el nodo solicitado es parte de un bosque y no de un solo árbol, o si el nodo antecesor tiene más de una representación normativa, debe haber al menos un ejemplar de este atributo para cada antecesor en el camino hacia las múltiples raíces de las que desciende el nodo solicitado, y para cada representación normativa de estos antecesores. El orden de los valores para este atributo no refleja necesariamente la posición de cada nodo antecesor en la jerarquía.

– Por cada antecesor del nodo especificado en el atributo o los atributos "resource-id", y para cada representación normativa de ese nodo antecesor, y por cada representación normativa del propio nodo "resource-id", un elemento <Attribute> que tenga como identificador (AttributeId) "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self". El <AttributeValue> de este <Attribute> debe ser la identidad normativa correspondiente del nodo antecesor o del propio nodo "resource-id". El DataType de este <Attribute> dependerá de la representación elegida para la identidad de los nodos en este recurso particular. Este <Attribute> puede especificar un Issuer. Para cada atributo "resource-ancestor" y "resource-id" habrá un atributo "resource-ancestor-or-self" correspondiente. Si el nodo solicitado es parte de un bosque y no de un solo árbol, o si el nodo antecesor tiene más de una representación normativa, debe haber al menos un ejemplar de este atributo para cada antecesor en el camino hacia las múltiples raíces de las que desciende el nodo solicitado, y para cada representación de los antecesores. El orden de los valores para este atributo no refleja necesariamente la posición para cada nodo progenitor en la jerarquía.

En el elemento <Resource> se pueden incluir atributos adicionales.

12.3 Establecimiento de políticas que se aplican a nodos

Esta cláusula es informativa.

Esta cláusula describe varias formas de especificar un predicado de política que se puede aplicar a múltiples nodos en un recurso jerárquico. No pretende ser una lista exhaustiva.

12.3.1 Políticas que se aplican a nodos en cualquier recurso jerárquico

Esta cláusula es informativa.

Los atributos de recurso con los siguientes valores de identificador (AttributeId), descritos en 12.5, se pueden utilizar para establecer políticas que se aplican a uno o varios nodos en cualquier recurso jerárquico.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-parent
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor-or-self
```

Hay que tener en cuenta que un <ResourceAttributeDesignator> que se refiere al atributo "resource-parent", "resource-ancestor", o "resource-ancestor-or-self" devolverá un multiconjunto de valores que representan todas las identidades normativas de todos los progenitores, los antecesores o la combinación de los antecesores y el propio recurso, respectivamente, del recurso al cual se ha solicitado acceso. Las representaciones de las identidades de estos progenitores, los antecesores o el mismo recurso no indican necesariamente el camino desde la raíz de la jerarquía hacia el respectivo progenitor, el antecesor o el mismo recurso, a no ser que se trate de la representación que se recomienda en 12.2.2.

Las funciones de los multiconjuntos XACML y de orden superior se pueden usar para establecer políticas que se aplican a uno o más nodos en cualquier recurso jerárquico. Los nodos usados como argumentos de estas funciones se pueden especificar usando un <ResourceAttributeDesignator> que tenga como valor de AttributeId de "resource-parent", "resource-ancestor", o "resource-ancestor-or-self".

12.3.2 Políticas que se aplican solamente a nodos en documentos XML

Esta cláusula es informativa.

Para los recursos jerárquicos que se representan como ejemplares de documentos XML, la siguiente función se puede utilizar para establecer predicados de política que se aplican a uno o más nodos en este recurso.

```
urn:oasis:names:tc:xacml:2.0:function:xpath-node-match
```

El elemento normal XACML `<AttributeSelector>` se puede utilizar en políticas para referirse a todos los fragmentos de un recurso representado como un documento XML y que se incluye en el elemento `<ResourceContent>` de un contexto de petición.

Las funciones de los multiconjuntos XACML y de orden superior se pueden usar para establecer políticas que se aplican a uno o varios nodos en un recurso representado como un documento XML. Los nodos usados como argumentos para estas funciones se pueden especificar usando un `<AttributeSelector>` que selecciona un fragmento del elemento `<ResourceContent>` del elemento `<Resource>`.

12.3.3 Políticas que se aplican solamente a nodos en documentos que no son XML

Esta cláusula es informativa.

Para los recursos jerárquicos que no se representan como ejemplares de documentos XML y en los que se usa la representación URI de nodos como se especifica en este perfil, las siguientes funciones se pueden usar para establecer políticas que se aplican a uno o más nodos en este recurso.

```
urn:oasis:names:tc:xacml:1.0:function:anyURI-equal
urn:oasis:names:tc:xacml:1.0:function:regexp-uri-match
```

12.4 Nuevo tipo de dato: expresión xpath

Esta cláusula es normativa, pero es facultativa.

La implementación puede soportar el siguiente valor para el valor del atributo XML `DataType`, para utilizarlo con recursos jerárquicos representados como documentos XML. Es necesario para soportar la cláusula 12.1.1.

El `DataType` representado por el siguiente URI representa una expresión XPath. Los valores de Atributos que tienen este `DataType` deben ser cadenas que serán interpretadas como expresiones XPath. El resultado de evaluar dichos atributos debe ser el grupo de nodos que resulta de evaluar la expresión XPath. Si la cadena no es una expresión XPath válida, el resultado de evaluar el atributo será "Indeterminado".

```
urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression
```

12.5 Nuevos identificadores de atributos

Esta cláusula es normativa, pero es facultativa.

12.5.1 document-id

El siguiente identificador indica la identidad del documento XML que representa la jerarquía a la que pertenece el recurso solicitado y que se ha copiado en el elemento `<ResourceContent>`. Cada vez que se solicita el acceso a un nodo en un recurso representado como un documento XML, se pueden proporcionar uno o varios ejemplares de atributos con este identificador en el elemento `<Resource>` del contexto de petición. El `DataType` de estos atributos será "`urn:oasis:names:tc:xacml:1.0:data-type:anyURI`".

```
urn:oasis:names:tc:xacml:2.0:resource:document-id
```

12.5.2 resource-parent

El siguiente identificador indica una identidad normativa de un nodo progenitor en el árbol o bosque del cual es parte el nodo solicitado. Cada vez que se solicita el acceso a un nodo en un recurso jerárquico, se proporcionará un ejemplar de un atributo con este identificador en el elemento `<Resource>` del contexto de petición para cada representación normativa de cada nodo progenitor del nodo solicitado.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-parent
```

12.5.3 resource-ancestor

El siguiente identificador indica una identidad normativa de un nodo antecesor en el árbol o bosque del cual es parte el nodo solicitado. Cada vez que se solicita el acceso a un nodo en un recurso jerárquico, se proporcionará un ejemplar de un atributo con este identificador en el elemento `<Resource>` del contexto de petición para cada representación normativa de cada nodo antecesor del nodo solicitado.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor
```

12.5.4 resource-ancestor-or-self

El siguiente identificador indica una identidad normativa de un nodo antecesor en el árbol o bosque del cual es parte el nodo solicitado, o una identidad normativa del propio nodo solicitado. Cada vez que se solicita el acceso a un nodo en un recurso jerárquico, se proporcionará un ejemplar de un atributo con este identificador en el elemento `<Resource>` del contexto de petición para cada representación normativa de cada nodo antecesor del nodo solicitado, y para cada representación normativa del propio nodo solicitado.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor-or-self
```

12.6 Nuevos identificadores de perfiles

Se utilizarán los siguientes valores de URI como identificadores para la funcionalidad especificada en varias cláusulas de este perfil:

Cláusula 12.1.1: Nodos en documentos XML

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-id
```

Cláusula 12.1.2: Nodos en recursos que no son documentos XML

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-id
```

Cláusula 12.2.1: Nodos en un documento XML

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req
```

En esta cláusula es facultativo soportar los atributos "`resource-parent`", "`resource-ancestor`", y "`resource-ancestor-or-self`"; por eso tienen identificadores separados:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-parent
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor-or-self
```

Cláusula 12.2.2: Nodos en un recurso que no es un documento XML

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req
```

En esta cláusula es facultativo soportar los atributos "`resource-parent`", "`resource-ancestor`", y "`resource-ancestor-or-self`"; por eso tienen identificadores separados:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-parent
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor-
or-self
```

13 Perfil de política de privacidad

Un custodio de datos tiene la obligación de garantizar que el uso de datos personales está limitado al cumplimiento de los propósitos para los que se registran, u otros propósitos no incompatibles con éstos, y prevenir la revelación de datos personales excepto con el consentimiento del sujeto de los datos o de la autoridad legal. Esta cláusula proporciona un perfil de atributos normales y un elemento `<Rule>` normal para cumplir estas obligaciones, basados en el propósito para el que se recaba y utiliza la información de tipo personal.

13.1 Atributos normales

Este perfil define dos atributos.

```
urn:oasis:names:tc:xacml:2.0:resource:purpose
```

Este atributo, del tipo "http://www.w3.org/2001/XMLSchema#string", indica el propósito de los datos. El propietario del recurso debe ser informado y consentir el uso del recurso para tal propósito. El valor del atributo puede ser una expresión regular. La política de privacidad del custodio debe definir la semántica de todos los valores disponibles.

```
urn:oasis:names:tc:xacml:2.0:action:purpose
```

Este atributo, del tipo "http://www.w3.org/2001/XMLSchema#string", indica el propósito para el que se solicita acceso al recurso de datos. Es posible organizar el propósito de forma jerárquica, en cuyo caso el valor debe representar un nodo en la jerarquía.

13.2 Reglas normales: Concordancia con el propósito

Esta regla debe utilizarse con el algoritmo combinador "`urn:oasis:names:tc:xacml:2.0:rule-combining-algorithm:deny-overrides`". Estipula que se denegará el acceso a menos que el propósito para el que se solicita acceso coincida (concordancia de expresiones) con el propósito para el que se ha registrado el recurso de datos.

```
<?xml version="1.0" encoding="UTF-8"?>
<Rule xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os" RuleId="
urn:oasis:names:tc:xacml:2.0:matching-purpose"
Effect="Permit">
      <Condition FunctionId="urn:oasis:names:tc:xacml:2.0:function:regexp-string-match">
            <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:2.0:resource:purpose"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
            <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:2.0:action:purpose"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
      </Condition>
</Rule>
```

Anexo A

Funciones y tipos de datos

A.1 Introducción

En este anexo se especifican las funciones y los tipos de datos utilizados en XACML con el fin de crear predicados para correspondencias de condiciones y metas.

En esta Recomendación se describen los tipos de datos y los multiconjuntos primitivos, se designan las funciones normalizadas y se describe su semántica operacional.

NOTA – Véase [IEEE 754] y [RBAC] para la representación de cadenas de información de valores numéricos.

A.2 Tipos de datos

Aunque los ejemplares XML representan todos los tipos de datos como cadenas, un PDP en XACML debe tratar con tipos de datos que son más que una cadena, aunque tengan representación de cadena. Los tipos de cadena, booleano, entero y doble deben convertirse. Las representaciones en cadenas XML se convertirán en valores que puedan compararse con valores en su dominio de discurso, tales como números. Se especifican los siguientes tipos de datos de primitivas para su utilización con XACML. Estos tipos de datos tienen representaciones explícitas.

- http://www.w3.org/2001/XMLSchema#string
- http://www.w3.org/2001/XMLSchema#boolean
- http://www.w3.org/2001/XMLSchema#integer
- http://www.w3.org/2001/XMLSchema#double
- http://www.w3.org/2001/XMLSchema#time
- http://www.w3.org/2001/XMLSchema#date
- http://www.w3.org/2001/XMLSchema#dateTime
- http://www.w3.org/2001/XMLSchema#anyURI
- http://www.w3.org/2001/XMLSchema#hexBinary
- http://www.w3.org/2001/XMLSchema#base64Binary
- urn:oasis:names:tc:xacml:2.0:data-type:dayTimeDuration
- urn:oasis:names:tc:xacml:2.0:data-type:yearMonthDuration
- urn:oasis:names:tc:xacml:1.0:data-type:x500Name
- urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name
- urn:oasis:names:tc:xacml:2.0:data-type:ipAddress
- urn:oasis:names:tc:xacml:2.0:data-type:dnsName

Para que haya más garantías de compatibilidad, se recomienda que todas las referencias de tiempo se expresen en hora universal (UTC).

Un PDP en XACML será capaz de convertir representaciones de cadenas en diversos tipos de datos primitivos.

NOTA – Para enteros y dobles, XACML utilizará las conversiones descritas en [IEEE 754].

XACML define seis tipos de datos, a saber:

```
"urn:oasis:names:tc:xacml:2.0:data-type:dayTimeDuration",
"urn:oasis:names:tc:xacml:2.0:data-type:yearMonthDuration",
"urn:oasis:names:tc:xacml:1.0:data-type:x500Name",
"urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
"urn:oasis:names:tc:xacml:2.0:data-type:ipAddress"
"urn:oasis:names:tc:xacml:2.0:data-type:dnsName"
```

Estos tipos representan identificadores para sujetos o recursos y aparecen en diversas aplicaciones normalizadas, tales como TLS/SSL y correo electrónico.

A.2.1 Duración en días y en tiempo

El tipo primitivo "urn:oasis:names:tc:xacml:2.0:data-type:dayTimeDuration" se define como una restricción del tipo **xs:duration**, conservando únicamente los componentes del signo, el año y el mes.

```
<xs:simpleType name='urn:oasis:names:tc:xacml:2.0:data-type:yearMonthDuration'>
    <xs:restriction base='xs:duration'>
        <xsd:pattern value="[-]?P\p{Nd}+(Y(\p{Nd}+M)?|M)"/>
    </xs:restriction>
</xs:simpleType>
```

El valor de una duración en años/meses (yearMonthDuration) se expresa en unidades de meses, y es:

```
 ('valor del componente año' * 12) + ('valor del componente mes')
```

Si el componente del signo es "-", el valor resultante es negativo; de otro modo, el valor resultante es positivo.

A.2.2 Duración en años y meses

El tipo primitivo "urn:oasis:names:tc:xacml:2.0:data-type:yearMonthDuration" se define como una restricción del tipo **xs:duration**, conservando únicamente los componentes de signo, día, hora, minuto y segundo.

```
<xs:simpleType name='urn:oasis:names:tc:xacml:2.0:data-type:dayTimeDuration'>
    <xs:restriction base='xs:duration'>
         <xsd:pattern value="[-
]?P(\p{Nd}D(T(\p{Nd}+(H(\p{Nd}+(M(\p{Nd}+(\.\p{Nd}*)?S
            |\.\p{Nd}+S)?|(\.\p{Nd}*)?S)|(\.\p{Nd}*)?S)?|M(\p{Nd}+
            (\.\p{Nd}*)?S|\.\p{Nd}+S)?|(\.\p{Nd}*)?S)|\.\p{Nd}+S))?
            |T(\p{Nd}+(H(\p{Nd}+(M(\p{Nd}+(\.\p{Nd}*)?S|\.\p{Nd}+S)?
            |(\.\p{Nd}*)?S)|(\.\p{Nd}*)?S)?|M(\p{Nd}+(\.\p{Nd}*)?S|\.\p{Nd}+S)?
            |(\.\p{Nd}*)?S)|\.\p{Nd}+S))"/>
    </xs:restriction>
</xs:simpleType>
```

El valor de una duración en tiempo (dayTimeDuration) se expresa en unidades de segundos, y es:

```
('value of the day component' * 24) +
('value of the hour component' * 60) +
('value of the minute component' * 60) +
('value of the second component')
```

Si el componente del signo es "-", el valor resultante es negativo; de otro modo el valor resultante es positivo.

A.2.3 Nombre del directorio X.500

El tipo primitivo "urn:oasis:names:tc:xacml:1.0:data-type:x500Name" representa un nombre distinguido X.520. La sintaxis válida para este tipo de nombre se describe en IETF RFC 2253.

A.2.4 Nombre conforme a IETF RFC 822

El tipo primitivo "urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name" representa una dirección de correo electrónico. La sintaxis válida para este tipo de nombre se describe en IETF RFC 2821, 4.1.2.

A.2.5 Dirección IP

El tipo primitivo "urn:oasis:names:tc:xacml:2.0:data-type:ipAddress" representa una dirección de red IPv4 ó IPv6, con máscara facultativa y puerto o gama de puertos facultativa. La sintaxis será:

```
ipAddress = address [ "/" mask ] [ ":" [ portrange ] ]
```

Para una dirección IPv4, la dirección y la máscara se formatean de conformidad con la sintaxis de un "anfitrión" en IETF RFC 2396, 3.2.

Para una dirección IPv6, la dirección y la máscara se formatean de conformidad con las sintaxis de una "ipv6reference" en IETF RFC 2732. (Obsérvese que una dirección o máscara IPv6, en esta sintaxis, está encerrada en corchetes "[" "]".)

A.2.6 Nombre de DNS

El tipo primitivo "urn:oasis:names:tc:xacml:2.0:data-type:dnsName" representa un nombre anfitrión del servicio de nombres de dominio (DNS, *domain name system*) con un puerto o una gama de puertos facultativa. La sintaxis será:

```
dnsName = hostname [ ":" portrange ]
```

El nombre del anfitrión se formatea de conformidad con IETF RFC 2396, 3.2, salvo que podría utilizarse un comodín "*" en el componente situado más a la izquierda del nombre anfitrión para indicar "cualquier subdominio" bajo el dominio especificado a su derecha.

Para los dos tipos de datos "`urn:oasis:names:tc:xacml:2.0:data-type:ipAddress`" y "`urn:oasis:names:tc:xacml:2.0:data-type:dnsName`", la sintaxis del puerto o del intervalo de puertos será:

```
portrange = portnumber | "-"portnumber | portnumber"-"[portnumber]
```

siendo "portnumber" un número de puerto decimal. Si el formato del número de puerto es "-x", donde "x" es un número de puerto, el intervalo será todos los puertos que tienen por número "x" o un número inferior. Si el formato es "x-", el intervalo será todos los puertos que tienen por número "x" o un número superior.

A.3 Funciones

XACML especifica las siguientes funciones. Si se determina que un argumento de una de estas funciones es "Indeterminate", se indicará que la función es "Indeterminate".

A.3.1 Predicados de igualdad

Las siguientes son las funciones de igualdad para los distintos tipos primitivos. Cada función obedece a una norma particular para un determinado tipo de datos.

`urn:oasis:names:tc:xacml:1.0:function:string-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#string" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los dos argumentos tienen valores de la misma longitud y se comprueba que las cadenas son iguales en todos los bytes, de conformidad con la función "integer-equal". Cuando no sea así, la respuesta será "Falso".

`urn:oasis:names:tc:xacml:1.0:function:boolean-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#boolean" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los argumentos son iguales. Cuando no sea así, la respuesta será "Falso".

`urn:oasis:names:tc:xacml:1.0:function:integer-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#integer" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean".

NOTA 1 – Véase [IEEE 754] para lo referente a la evaluación de enteros.

`urn:oasis:names:tc:xacml:1.0:function:double-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#double" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean".

NOTA 2 – Véase [IEEE 754] para lo referente a la evaluación de dobles.

`urn:oasis:names:tc:xacml:1.0:function:date-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#date" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los dos argumentos tienen el mismo valor. Si en uno de los argumentos no hay una indicación explícita del huso horario, es preciso indicarla en la implementación.

`urn:oasis:names:tc:xacml:1.0:function:time-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#time" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los dos argumentos tienen el mismo valor. Si en uno de los argumentos no hay una indicación explícita del huso horario, es preciso indicarla en la implementación.

`urn:oasis:names:tc:xacml:1.0:function:dateTime-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#dateTime" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los dos argumentos tienen el mismo valor. Si en uno de los argumentos no hay una indicación explícita del huso horario, es preciso indicarla en la implementación.

`urn:oasis:names:tc:xacml:1.0:function:dayTimeDuration-equal`

Esta función tendrá dos argumentos de datos del tipo "`urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration`" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los dos argumentos tienen el mismo valor. Obsérvese que es necesario convertir la representación léxica de cada argumento en un valor de fracciones de segundo.

`urn:oasis:names:tc:xacml:1.0:function:yearMonthDuration-equal`

Esta función tendrá dos argumentos de datos del tipo "`urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration`" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los dos argumentos tienen el mismo valor. Obsérvese que es necesario convertir la representación léxica de cada argumento en un valor entero de meses.

`urn:oasis:names:tc:xacml:1.0:function:anyURI-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#anyURI" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los dos argumentos tienen el mismo valor para cada punto de código.

`urn:oasis:names:tc:xacml:1.0:function:x500Name-equal`

Esta función tendrá dos argumentos de "`urn:oasis:names:tc:xacml:1.0:data-type:x500Name`" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si coinciden todos los nombres distinguidos relativos (RDN, *relative distinguished name*) de los dos argumentos. Cuando no sea así, la respuesta será "Falso". Se considerará que dos nombres RDN coinciden sólo si el resultado de las siguientes operaciones es "Verdadero".

1) Normalizar los dos argumentos conforme a IETF RFC 2253.
2) Si un RDN contiene múltiples pares `attributeTypeAndValue`, reordenar los `AttributeValuePairs` de ese RDN en orden ascendente, considerándolos como cadenas de octetos (véase 11.6/X.690).
3) Comparar los RDN conforme a las reglas de IETF RFC 3280, 4.1.2.4.

`urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal`

Esta función tendrá dos argumentos de datos del tipo "`urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name`" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" únicamente si los dos argumentos son iguales. Cuando no sea así, la respuesta será "Falso". Los nombres IETF RFC 822 están formados por una parte local seguida de "@" y una parte de dominio. La parte local es sensible a mayúsculas-minúsculas, pero no la parte de dominio (que generalmente es un nombre de anfitrión DNS). Se hacen las siguientes operaciones:

1) Normalizar la parte de dominio de cada argumento en minúsculas.
2) Comparar las expresiones aplicando la función "`urn:oasis:names:tc:xacml:1.0:function:string-equal`" a los argumentos normalizados.

`urn:oasis:names:tc:xacml:1.0:function:hexBinary-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#hexBinary" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" si las secuencias de octetos representadas por el valor de los dos argumentos tienen la misma longitud, y se determina que son iguales en una comparación conjuntiva y por puntos, utilizando la función "`urn:oasis:names:tc:xacml:1.0:function:integer-equal`". Cuando no sea así, la respuesta será "Falso". La conversión de la representación de cadena en secuencia de octetos se hará según la especificación de W3C Datatypes:2001, 3.2.15.

`urn:oasis:names:tc:xacml:1.0:function:base64Binary-equal`

Esta función tendrá dos argumentos de datos del tipo "http://www.w3.org/2001/XMLSchema#base64Binary" y devolverá un "http://www.w3.org/2001/XMLSchema#boolean". La respuesta de la función será "Verdadero" si las secuencias de octetos representadas por el valor de los dos argumentos tienen la misma longitud, y se determina que son

iguales en una comparación conjuntiva y por puntos, utilizando la función "`urn:oasis:names:tc:xacml:1.0:function:integer-equal`". Cuando no sea así, la respuesta será "Falso". La conversión de la representación de cadena en secuencia de octetos se hará según la especificación de W3C Datatypes:2001, 3.2.16.

A.3.2 Funciones aritméticas

Todas las siguientes funciones tendrán dos argumentos del tipo de datos especificado, enteros o dobles, y originarán un elemento de tipo de datos entero o doble, respectivamente. Ahora bien, las funciones "add" pueden tener más de dos argumentos. Si hubiera un argumento "Indeterminate" en una expresión que contiene una de estas funciones, se establecerá que la expresión es "Indeterminate". En el caso de las funciones de división, si el divisor es cero, el resultado de evaluación de la expresión será "Indeterminate".

NOTA – La evaluación de cada función se hará como especifican las contrapartes lógicas en [IEEE 754].

```
urn:oasis:names:tc:xacml:1.0:function:integer-add
```

Esta función puede tener dos o más argumentos.

```
urn:oasis:names:tc:xacml:1.0:function:double-add
```

Esta función puede tener dos o más argumentos.

```
urn:oasis:names:tc:xacml:1.0:function:integer-subtract
urn:oasis:names:tc:xacml:1.0:function:double-subtract
urn:oasis:names:tc:xacml:1.0:function:integer-multiply
urn:oasis:names:tc:xacml:1.0:function:double-multiply
urn:oasis:names:tc:xacml:1.0:function:integer-divide
urn:oasis:names:tc:xacml:1.0:function:double-divide
urn:oasis:names:tc:xacml:1.0:function:integer-mod
```

Las siguientes funciones tendrán un solo argumento del tipo de datos especificado. Las funciones de redondeo y redondeo al entero inferior tienen un solo argumento del tipo de datos "http://www.w3.org/2001/XMLSchema#double" y devolverán un valor del tipo de datos "http://www.w3.org/2001/XMLSchema#double".

```
urn:oasis:names:tc:xacml:1.0:function:integer-abs
urn:oasis:names:tc:xacml:1.0:function:double-abs
urn:oasis:names:tc:xacml:1.0:function:round
urn:oasis:names:tc:xacml:1.0:function:floor
```

A.3.3 Funciones de conversión de cadenas de valores

Las siguientes funciones realizan la conversión de valores de tipos de datos primitivos "http://www.w3.org/2001/XMLSchema#string".

```
urn:oasis:names:tc:xacml:1.0:function:string-normalize-space
```

Esta función se aplica a un argumento del tipo de datos "http://www.w3.org/2001/XMLSchema#string" para normalizar el valor retirando los espacios en blanco al principio y al final.

```
urn:oasis:names:tc:xacml:1.0:function:string-normalize-to-lower-case
```

Esta función se aplica a un argumento del tipo de datos "http://www.w3.org/2001/XMLSchema#string" para normalizar el valor convirtiendo cada carácter en mayúscula en su equivalente en minúscula.

A.3.4 Funciones de conversión de tipos de datos numéricos

Las siguientes funciones realizan la conversión entre los tipos de datos primitivos "http://www.w3.org/2001/XMLSchema#integer" y "http://www.w3.org/2001/XMLSchema#double".

```
urn:oasis:names:tc:xacml:1.0:function:double-to-integer
```

Esta función se aplica a un argumento del tipo de datos "http://www.w3.org/2001/XMLSchema#double" para truncar el valor numérico hasta el entero y devuelve un elemento del tipo de datos "http://www.w3.org/2001/XMLSchema#integer".

```
urn:oasis:names:tc:xacml:1.0:function:integer-to-double
```

Esta función se aplica a un argumento del tipo de datos "http://www.w3.org/2001/XMLSchema#integer" y convertirá su valor en un elemento del tipo de datos "http://www.w3.org/2001/XMLSchema#double" con el mismo valor numérico.

A.3.5 Funciones lógicas

En esta cláusula se especifican las funciones lógicas que modifican argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#boolean".

```
urn:oasis:names:tc:xacml:1.0:function:or
```

La respuesta de esta función será "Falso" si no tiene argumentos o "Verdadero" si el resultado de evaluación de uno o más de sus argumentos es "Verdadero". La evaluación empieza por el primer argumento, hasta el último. Se suspenderá cuando el resultado de evaluación de un argumento sea "Verdadero" y no se evaluarán los demás argumentos.

```
urn:oasis:names:tc:xacml:1.0:function:and
```

La respuesta de esta función será "Verdadero" si no tiene argumentos o "Falso" si el resultado de evaluación de uno o más de sus argumentos es "Falso". La evaluación empieza por el primer argumento, hasta el último. Se suspenderá cuando el resultado de evaluación de un argumento sea "Falso" y no se evaluarán los demás argumentos.

```
urn:oasis:names:tc:xacml:1.0:function:n-of
```

El primer argumento de esta función será del tipo de datos "http://www.w3.org/2001/XMLSchema#integer". Los demás argumentos serán del tipo de datos "http://www.w3.org/2001/XMLSchema#boolean". El primer argumento especifica el número mínimo de otros argumentos que tendrán que dar un resultado de evaluación "Verdadero" para que se considere que el resultado de evaluación de la expresión es "Verdadero". Si el primer argumento es 0, el resultado será "Verdadero". Si hay un número de argumentos después del primero inferior al valor que indica el primer argumento, el resultado de evaluación de la expresión será "Indeterminate". La evaluación se hará en este orden: primero el valor entero y después cada uno de los argumentos siguientes. La evaluación se suspenderá y el resultado será "Verdadero" si el resultado de evaluación del número de argumentos especificado es "Verdadero". La evaluación de argumentos se suspenderá si se determina que la evaluación de los demás argumentos no permitirá cumplir la condición.

```
urn:oasis:names:tc:xacml:1.0:function:not
```

Esta función se aplica a un argumento del tipo de datos "http://www.w3.org/2001/XMLSchema#boolean". Si el resultado de evaluación del argumento es "Verdadero", el resultado de la expresión será "Falso". Si el resultado de evaluación del argumento es "Falso", el resultado de la expresión será "Verdadero".

NOTA – Cuando se hace una evaluación basada en los criterios "y", "o" y "n-de", no siempre habrá que hacer una evaluación completa de cada argumento para determinar si el resultado de evaluación del argumento será "Indeterminate". Es posible eliminar la posibilidad de errores del argumento haciendo un análisis de disponibilidad de atributos y otros análisis de errores, por ejemplo "dividir por cero". No es necesario procesar estos argumentos cuando aparecen en la expresión más allá del punto en que se ha de suspender la evaluación.

A.3.6 Funciones de comparaciones numéricas

Estas funciones constituyen un conjunto mínimo para comparar dos números y producen un resultado booleano.

NOTA – Estas funciones deben ajustarse a las normas de [IEEE 754].

```
urn:oasis:names:tc:xacml:1.0:function:integer-greater-than
urn:oasis:names:tc:xacml:1.0:function:integer-greater-than-or-equal
urn:oasis:names:tc:xacml:1.0:function:integer-less-than
urn:oasis:names:tc:xacml:1.0:function:integer-less-than-or-equal
urn:oasis:names:tc:xacml:1.0:function:double-greater-than
urn:oasis:names:tc:xacml:1.0:function:double-greater-than-or-equal
urn:oasis:names:tc:xacml:1.0:function:double-less-than
urn:oasis:names:tc:xacml:1.0:function:double-less-than-or-equal
```

A.3.7 Funciones aritméticas de fecha y hora

Estas funciones realizan operaciones aritméticas con la fecha y la hora.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-add-dayTimeDuration
```

Esta función se aplica a dos argumentos, uno del tipo de datos "http://www.w3.org/2001/XMLSchema#dateTime" y el otro del tipo de datos "`urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration`". El resultado será "http://www.w3.org/2001/XMLSchema#dateTime". Esta función suma el segundo argumento al primero para producir el resultado, conforme a W3C DataTypes:2001, apéndice E.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-add-yearMonthDuration
```

Esta función se aplica a dos argumentos: uno será "http://www.w3.org/2001/XMLSchema#dateTime" y el otro será "urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration". El resultado será "http://www.w3.org/2001/XMLSchema#dateTime". Esta función suma el segundo argumento al primero para producir el resultado, conforme a W3C DataTypes:2001, apéndice E.

urn:oasis:names:tc:xacml:1.0:function:dateTime-subtract-dayTimeDuration

Esta función se aplica a dos argumentos: uno será "http://www.w3.org/2001/XMLSchema#dateTime" y el otro será "urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration". El resultado será "http://www.w3.org/2001/XMLSchema#dateTime". Si el segundo argumento es una duración positiva, esta función producirá el valor sumando la duración negativa correspondiente conforme a W3C DataTypes:2001, apéndice E. Si el segundo argumento es una duración negativa, se obtendrá el mismo resultado que se habría obtenido aplicando la función "urn:oasis:names:tc:xacml:1.0:function:dateTime-add-dayTimeDuration" a la duración positiva correspondiente.

urn:oasis:names:tc:xacml:1.0:function:dateTime-subtract-yearMonthDuration

Esta función se aplica a dos argumentos: uno será "http://www.w3.org/2001/XMLSchema#dateTime" y el otro será "urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration". El resultado será "http://www.w3.org/2001/XMLSchema#dateTime". Si el segundo argumento es una duración positiva, esta función producirá el valor sumando la duración negativa correspondiente conforme a W3C DataTypes:2001, apéndice E. Si el segundo argumento es una duración negativa, se obtendrá el mismo resultado que se habría obtenido aplicando la función "urn:oasis:names:tc:xacml:1.0:function:dateTime-add-yearMonthDuration" a la duración positiva correspondiente.

urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration

Esta función se aplica a dos argumentos: uno será "http://www.w3.org/2001/XMLSchema#date" y el otro será "urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration". El resultado será "http://www.w3.org/2001/XMLSchema#date". Esta función suma el segundo argumento al primero para producir el resultado, conforme a W3C DataTypes:2001, apéndice E.

urn:oasis:names:tc:xacml:1.0:function:date-subtract-yearMonthDuration

Esta función se aplica a dos argumentos: uno será "http://www.w3.org/2001/XMLSchema#date" y el otro será "urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration". El resultado será "http://www.w3.org/2001/XMLSchema#date". Si el segundo argumento es una duración positiva, esta función producirá el valor sumando la duración negativa correspondiente conforme a W3C DataTypes:2001, apéndice E. Si el segundo argumento es una duración negativa, se obtendrá el mismo resultado que se habría obtenido aplicando la función "urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration" a la duración positiva correspondiente.

A.3.8 Funciones de comparación no numéricas

Estas funciones hacen operaciones de comparación de dos argumentos no numéricos.

urn:oasis:names:tc:xacml:1.0:function:string-greater-than

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#string" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si los argumentos se comparan byte por byte y se comprueba lo siguiente: después del prefijo inicial de bytes correspondientes de los dos argumentos, que se consideran iguales en una comparación mediante "urn:oasis:names:tc:xacml:1.0:function:integer-equal", en los demás bytes se comprueba que el byte del primer argumento es mayor que el byte del segundo, comparados mediante la función "urn:oasis:names:tc:xacml:2.0:function:integer-greater-then". Cuando no sea así, el resultado será "Falso".

urn:oasis:names:tc:xacml:1.0:function:string-greater-than-or-equal

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#string" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será el mismo que se habría obtenido con la función lógica "urn:oasis:names:tc:xacml:1.0:function:or" con dos argumentos que contienen las funciones "urn:oasis:names:tc:xacml:1.0:function:string-greater-than" y "urn:oasis:names:tc:xacml:1.0:function:string-equal" y los argumentos originales.

urn:oasis:names:tc:xacml:1.0:function:string-less-than

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#string" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si los argumentos se comparan byte por byte y se comprueba lo siguiente: después del prefijo inicial de bytes correspondientes de los dos argumentos, que se consideran iguales en una comparación mediante "`urn:oasis:names:tc:xacml:1.0:function:integer-equal`", en los demás bytes se comprueba que el byte del primer argumento es menor que el byte del segundo, comparados mediante la función "`urn:oasis:names:tc:xacml:1.0:function:integer-less-than`". Cuando no sea así, el resultado será "Falso".

`urn:oasis:names:tc:xacml:1.0:function:string-less-than-or-equal`

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#string" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será el mismo que se habría obtenido evaluando con la función "`urn:oasis:names:tc:xacml:1.0:function:or`" con dos argumentos que contienen las funciones "`urn:oasis:names:tc:xacml:1.0:function:string-less-than`" y "`urn:oasis:names:tc:xacml:1.0:function:string-equal`" y los argumentos originales.

`urn:oasis:names:tc:xacml:1.0:function:time-greater-than`

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#time" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es mayor que el segundo, conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#time" (W3C Signature:2002, 3.2.8). Cuando no sea así, el resultado será "Falso".

NOTA 1 – No es apropiado comparar una hora que tiene indicación de huso horario con otra que no la tiene. En estos casos se ha de utilizar la función "*time-in-range*" (hora entre dos valores).

`urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal`

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#time" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es mayor que el segundo, o igual, conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#time" (W3C Datatypes:2001, 3.2.8). Cuando no sea así, el resultado será "Falso".

NOTA 2 – No es apropiado comparar una hora que tiene indicación de huso horario con otra que no la tiene. En estos casos se ha de utilizar la función "*time-in-range*" (hora entre dos valores).

`urn:oasis:names:tc:xacml:1.0:function:time-less-than`

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#time" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es menor que el segundo, conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#time" (W3C Datatypes:2001, 3.2.8). Cuando no sea así, el resultado será "Falso".

NOTA 3 – No es apropiado comparar una hora que tiene indicación de huso horario con otra que no la tiene. En estos casos se ha de utilizar la función "*time-in-range*" (hora entre dos valores).

`urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal`

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#time" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es menor que el segundo, o igual, conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#time". Cuando no sea así, el resultado será "Falso".

NOTA 4 – No es apropiado comparar una hora que tiene indicación de huso horario con otra que no la tiene. En estos casos se ha de utilizar la función "*time-in-range*" (hora entre dos valores).

`urn:oasis:names:tc:xacml:1.0:function:time-in-range`

Esta función se aplica a tres argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#time" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento está dentro de los límites especificados por el segundo y el tercer argumentos, incluidos sus valores. Cuando no sea así, el resultado será "Falso". Independientemente de su valor, el tercer argumento será considerado como una hora igual al segundo argumento, o posterior en menos de 24 horas. Cuando no se indique el huso horario para el primer argumento, se utilizará el huso por defecto de la función de servicio. Cuando no se indique el huso horario para el segundo y el tercer argumentos, se utilizará el huso del primero.

`urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than`

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#dateTime" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es mayor que el segundo conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#dateTime" por W3C Datatype:2001, 3.2.7. Cuando no sea así, el resultado será "Falso".

NOTA 5 – Si un valor de `dateTime` no tiene indicación de huso horario, se asignará un valor implícito de huso horario, conforme a la descripción de W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-greater-than-or-equal
```

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#dateTime" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es mayor que el segundo, o igual, conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#dateTime" por W3C Datatype:2001, 3.2.7. Cuando no sea así, el resultado será "Falso".

NOTA 6 – Si un valor de `dateTime` no tiene indicación de huso horario, se asignará un valor implícito de huso horario, conforme a la descripción de W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than
```

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#dateTime" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es menor que el segundo conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#dateTime" por W3C Datatype:2001, 3.2.7. Cuando no sea así, el resultado será "Falso".

NOTA 7 – Si un valor de `dateTime` no tiene indicación de huso horario, se asignará un valor implícito de huso horario, conforme a la descripción de W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:dateTime-less-than-or-equal
```

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema# dateTime" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es menor que el segundo, o igual, conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#dateTime" por W3C Datatypes:2001, 3.2.7. Cuando no sea así, el resultado será "Falso".

NOTA 8 – Si un valor de `dateTime` no tiene indicación de huso horario, se asignará un valor implícito de huso horario, conforme a la descripción de W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:date-greater-than
```

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#date" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es mayor que el segundo conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#date". Cuando no sea así, el resultado será "Falso".

NOTA 9 – Si un valor de fecha no tiene indicación de huso horario, se asignará un valor implícito de huso horario, conforme a la descripción de W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:date-greater-than-or-equal
```

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#date" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es mayor que el segundo, o igual, conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#date". Cuando no sea así, el resultado será "Falso".

NOTA 10 – Si un valor de fecha no tiene indicación de huso horario, se asignará un valor implícito de huso horario, conforme a la descripción de W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:date-less-than
```

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#date" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es menor que el segundo conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#date". Cuando no sea así, el resultado será "Falso".

NOTA 11 – Si un valor de fecha no tiene indicación de huso horario, se asignará un valor implícito de huso horario, conforme a la descripción de W3C Datatype:2001.

```
urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal
```

Esta función se aplica a dos argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#date" y dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado será "Verdadero" si el primer argumento es menor o igual que el segundo conforme a la relación de orden especificada para "http://www.w3.org/2001/XMLSchema#date". Cuando no sea así, el resultado será "Falso".

NOTA 12 – Si un valor de fecha no tiene indicación de huso horario, se asignará un valor implícito de huso horario, conforme a la descripción de W3C Datatype:2001.

A.3.9 Funciones para cadenas de caracteres

Las siguientes funciones se aplican a cadenas de caracteres y URI.

```
urn:oasis:names:tc:xacml:2.0:function:string-concatenate
```

Esta función se aplica a dos o más argumentos del tipo de datos "http://www.w3.org/2001/XMLSchema#string" y dará como resultado un "http://www.w3.org/2001/XMLSchema#string". El resultado será una concatenación, ordenada, de los argumentos.

```
urn:oasis:names:tc:xacml:2.0:function:url-string-concatenate
```

Esta función se aplica a un argumento del tipo de datos "http://www.w3.org/2001/XMLSchema#anyURI" y uno o más argumentos del tipo "http://www.w3.org/2001/XMLSchema#string", y dará como resultado un "http://www.w3.org/2001/XMLSchema#anyURI". El resultado será el URI que se forma anexando, en orden, los argumentos "cadena" al argumento "any URI".

A.3.10 Funciones de multiconjuntos

Estas funciones se aplican a un multiconjunto de valores de 'tipo' (uno de los tipos de datos primitivos). Hay otras condiciones que se definen para cada una de las siguientes funciones y que harán que el resultado de la evaluación de la expresión sea "Indeterminate".

```
urn:oasis:names:tc:xacml:1.0:function:type-one-and-only
```

Esta función se aplica a un multiconjunto de valores de 'tipo' como argumento y el resultado es un valor relativo al '-tipo'. El resultado será el único valor del multiconjunto. Si en el multiconjunto hubiera más de un valor, el resultado de la evaluación de la expresión sería "Indeterminate".

```
urn:oasis:names:tc:xacml:1.0:function:type-bag-size
```

Esta función se aplica a un multiconjunto de valores de 'tipo' como argumento y el resultado es un "http://www.w3.org/2001/XMLSchema#integer" que indica el número de valores del multiconjunto.

```
urn:oasis:names:tc:xacml:1.0:function:type-is-in
```

Esta función se aplica a dos argumentos: el primero es 'tipo' y el segundo es un multiconjunto de valores de 'tipo'. Dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado de evaluación de esta función será "Verdadero" si el primer argumento coincide con alguno de los valores del multiconjunto conforme a "`urn:oasis:names:tc:xacml:x.x:function:type-equal`". Cuando no sea así, el resultado será "Falso".

```
urn:oasis:names:tc:xacml:1.0:function:type-bag
```

Esta función se aplica a cualquier número de argumentos de 'tipo' y dará como resultado un multiconjunto de valores de 'tipo' que contiene los valores de los argumentos. Si esta función se aplica a cero argumentos, el resultado será un multiconjunto vacío del tipo de datos especificado.

A.3.11 Funciones de conjuntos

Estas funciones se aplican a multiconjuntos para convertirlos en conjuntos suprimiendo los elementos duplicados.

```
urn:oasis:names:tc:xacml:1.0:function:type-intersection
```

Esta función se aplica a dos argumentos, siendo ambos multiconjuntos de valores de 'tipo'. El resultado será un multiconjunto de valores de 'tipo' que sólo contiene elementos comunes a los dos multiconjuntos, lo que se determina mediante "`urn:oasis:names:tc:xacml:x.x:function:type-equal`". En el resultado no habrá duplicados, lo que se determina mediante "`urn:oasis:names:tc:xacml:x.x:function:type-equal`".

```
urn:oasis:names:tc:xacml:1.0:function:type-at-least-one-member-of
```

Esta función se aplica a dos argumentos, siendo ambos multiconjuntos de valores de 'tipo'. El resultado será un "http://www.w3.org/2001/XMLSchema#boolean". El resultado de evaluación de la función será "Verdadero" si hay al menos un elemento del primer argumento incluido en el segundo argumento, lo que se determina mediante "urn:oasis:names:tc:xacml:x.x:function:type-is-in".

urn:oasis:names:tc:xacml:1.0:function:type-union

Esta función se aplica a dos argumentos, siendo ambos multiconjuntos de valores de 'tipo'. El resultado será un multiconjunto de valores de "tipo" que contiene todos los elementos de los dos multiconjuntos. En el resultado no habrá duplicados, lo que se determina mediante "urn:oasis:names:tc:xacml:x.x:function:type-equal".

urn:oasis:names:tc:xacml:1.0:function:type-subset

Esta función se aplica a dos argumentos, siendo ambos multiconjuntos de valores de 'tipo'. El resultado será un "http://www.w3.org/2001/XMLSchema#boolean". El resultado de evaluación de la función será "Verdadero" si el primer argumento es un subconjunto del segundo. Se considerará que se han suprimido los duplicados de los dos argumentos, lo que se determina mediante "urn:oasis:names:tc:xacml:x.x:function:type-equal", antes de calcular el subconjunto.

urn:oasis:names:tc:xacml:1.0:function:type-set-equals

Esta función se aplica a dos argumentos, siendo ambos multiconjuntos de valores de 'tipo'. Dará como resultado un "http://www.w3.org/2001/XMLSchema#boolean". El resultado es el que se obtiene con la función "urn:oasis:names:tc:xacml:1.0:function:and" en la aplicación de "urn:oasis:names:tc:xacml:x.x:function:type-subset" al primero y el segundo argumentos, y en la aplicación de "urn:oasis:names:tc:xacml:x.x:function:type-subset" al segundo y el primer argumentos.

A.3.12 Funciones de multiconjuntos de orden superior

Esta cláusula está dedicada a las funciones en XACML que realizan operaciones en multiconjuntos y que pueden aplicarse a los multiconjuntos en general.

Sería útil disponer de un lenguaje funcional general para especificar la semántica de estas funciones (véase un ejemplo informativo de utilización de un lenguaje funcional en el apéndice III).

1) urn:oasis:names:tc:xacml:1.0:function:any-of

   Esta función aplica una función booleana entre un determinado valor primitivo y un multiconjunto de valores. El resultado será "Verdadero" si el resultado de evaluación del predicado es "Verdadero" al menos para un elemento del multiconjunto.

   Esta función se aplica a tres argumentos. El primer argumento será un elemento <xacml:Function> que designa una función booleana aplicada a dos argumentos de tipos primitivos. El segundo argumento será un valor de un tipo de datos primitivo. El tercer argumento será un multiconjunto de un tipo de datos primitivo. La evaluación de la expresión consiste en aplicar la función denominada en el argumento <xacml:Function> al segundo argumento y todos los elementos del tercer argumento (multiconjunto), y combinar los resultados con "urn:oasis:names:tc:xacml:1.0:function:or".

2) urn:oasis:names:tc:xacml:1.0:function:all-of

   Esta función aplica una función booleana entre un determinado valor primitivo y un multiconjunto de valores. El resultado será "Verdadero" si el resultado de evaluación del predicado es "Verdadero" para todos los elementos del multiconjunto.

   Esta función se aplica a tres argumentos. El primer argumento será un elemento <xacml:Function> que designa una función booleana aplicada a dos argumentos de tipos primitivos. El segundo argumento será un valor de un tipo de datos primitivo. El tercer argumento será un multiconjunto de un tipo de datos primitivo. La evaluación de la expresión consiste en aplicar la función denominada en el argumento <xacml:Function> al segundo argumento y todos los elementos del tercer argumento (multiconjunto), y combinar los resultados con "urn:oasis:names:tc:xacml:1.0:function:and".

3) urn:oasis:names:tc:xacml:1.0:function:any-of-any

   Esta función aplica una función booleana entre cada elemento de un multiconjunto de valores y cada elemento de otro multiconjunto de valores. Dará como resultado "Verdadero" si el resultado de evaluación del predicado es "Verdadero" al menos para una de las comparaciones.

   Esta función se aplica a tres argumentos. El primer argumento será un elemento <xacml:Function> que designa una función booleana aplicada a dos argumentos de tipos primitivos. El segundo argumento será un multiconjunto de un tipo de datos primitivo. El tercer argumento también será un multiconjunto

de un tipo de datos primitivo. La evaluación de la expresión consiste en aplicar la función denominada en el argumento <xacml:Function> entre todos los elementos del segundo argumento y todos los elementos del tercer argumento, y combinar los resultados con "urn:oasis:names:tc:xacml:1.0:function:or". Con esta semántica, el resultado de evaluación de la expresión será "Verdadero" si el predicado aplicado es "Verdadero" al menos para una de las comparaciones entre elementos de los dos multiconjuntos.

4) urn:oasis:names:tc:xacml:1.0:function:all-of-any

Esta función aplica una función booleana entre los elementos de dos multiconjuntos. El resultado de evaluación de la expresión será "Verdadero" sólo si el predicado aplicado es "Verdadero" entre cada uno de los elementos del primer multiconjunto y cualquiera de los elementos del segundo.

Esta función se aplica a tres argumentos. El primer argumento será un elemento <xacml:Function> que designa una función booleana aplicada a dos argumentos de tipos primitivos. El segundo argumento será un multiconjunto de un tipo de datos primitivo. El tercer argumento también será un multiconjunto de un tipo de datos primitivo. La evaluación de la expresión consiste en aplicar la función "urn:oasis:names:tc:xacml:1.0:function:any-of" a cada uno de los valores del primer multiconjunto y la totalidad del segundo, utilizando la función xacml:Function proporcionada, y combinar los resultados con "urn:oasis:names:tc:xacml:1.0:function:and".

5) urn:oasis:names:tc:xacml:1.0:function:any-of-all

Esta función aplica una función booleana entre los elementos de dos multiconjuntos. El resultado de evaluación de la expresión será "Verdadero" si el predicado aplicado es "Verdadero" entre cada uno de los elementos del segundo multiconjunto y cualquiera de los elementos del primero.

Esta función se aplica a tres argumentos. El primer argumento será un elemento <xacml:Function> que designa una función booleana aplicada a dos argumentos de tipos primitivos. El segundo argumento será un multiconjunto de un tipo de datos primitivo. El tercer argumento también será un multiconjunto de un tipo de datos primitivo. La evaluación de la expresión consiste en aplicar la función "urn:oasis:names:tc:xacml:1.0:function:any-of" a cada uno de los valores del segundo multiconjunto y la totalidad del primero, utilizando la función xacml:Function proporcionada, y combinar los resultados con "urn:oasis:names:tc:xacml:1.0:function:and".

6) urn:oasis:names:tc:xacml:1.0:function:all-of-all

Esta función aplica una función booleana entre los elementos de dos multiconjuntos. El resultado de la expresión será "Verdadero" si y sólo si el predicado suministrado es "Verdadero" para todos y cada uno de los elementos del primer multiconjunto contrastados colectivamente con todos los elementos del segundo.

Esta función tiene tres argumentos. El primer argumento es un elemento <xacml:Function> que designa una función booleana aplicada a dos argumentos de tipos primitivos. El segundo argumento es un multiconjunto de un tipo de datos primitivo. El tercer argumento también es un multiconjunto de un tipo de datos primitivo. La evaluación de la expresión consiste en aplicar la función designada en el elemento <xacml:Function> entre cada elemento del segundo argumento y cada elemento del tercer argumento, y combinar los resultados utilizando "urn:oasis:names:tc:xacml:1.0:function:and". Con esta semántica, el resultado de la expresión será "Verdadero" si y sólo si el predicado que se aplica es "Verdadero" para todos los elementos del primer multiconjunto comparados con todos los elementos del segundo.

7) urn:oasis:names:tc:xacml:1.0:function:map

Esta función convierte un multiconjunto de valores en otro.

La función tiene dos argumentos. El primero es un elemento <xacml:Function> que designa una función aplicada a un argumento único de un tipo de dato primitivo y devuelve un valor de un tipo de dato primitivo. El segundo argumento es un multiconjunto de un tipo de datos primitivo. La evaluación de la expresión consiste en aplicar la función designada en el elemento <xacml:Function> a cada elemento del multiconjunto, para producir otro multiconjunto de otro valor. El resultado será un multiconjunto del tipo de datos primitivo, devuelto por la función designada en el elemento <xacml:Function>.

A.3.13 Funciones basadas en expresiones regulares

Estas funciones operan sobre varios tipos, con expresiones regulares y el resultado de evaluación es "http://www.w3.org/2001/XMLSchema#boolean".

```
urn:oasis:names:tc:xacml:1.0:function:string-regexp-match
```

Esta función determina si hay concordancia de una expresión regular. Tiene dos argumentos "http://www.w3.org/2001/XMLSchema#string" y devuelve un "http://www.w3.org/2001/XMLSchema#boolean". El primer argumento es una expresión regular, y el segundo argumento, una cadena general. La función devolverá "Verdadero" si y sólo si el segundo argumento concuerda con el valor del modelo de la expresión regular del primer argumento. La sintaxis de las expresiones regulares es la que se define en W3C Datatypes:2001, apéndice F, y además sigue las siguientes reglas y expresiones adicionales del modelo:

- X?? será concordante con X no más de una vez
- X*? será concordante con X en todas las ocasiones, incluido el cero
- X+? será concordante con X al menos una vez
- X{n}? será concordante con X exactamente n veces
- X(n,}? será concordante con X al menos n veces
- X{n,m}? será concordante con X al menos n veces, pero no más de m veces

Cuando se utilice una de esas expresiones adicionales del modelo, la expresión regular será concordante con la subcadena del primer argumento más corta posible que se ajuste al modelo.

Excepto en el caso de esas expresiones adicionales, la expresión regular será concordante con la subcadena del primer argumento más larga posible que se ajuste al modelo.

Excepto cuando explícitamente se vincule al principio o fin de la cadena utilizando los caracteres del modelo "^" o "$", respectivamente, se considerará que el modelo concuerda si hay concordancia con cualquier subcadena del segundo argumento.

Todas las implementaciones conformes tienen que soportar los modelos especificados de expresiones regulares. Una implementación conforme puede soportar otros modelos de expresiones regulares.

```
urn:oasis:names:tc:xacml:2.0:function:anyURI-regexp-match
```

Esta función determina si hay concordancia de una expresión regular. Tiene dos argumentos, el primero de los cuales es de tipo "http://www.w3.org/2001/XMLSchema#string" y el segundo de tipo "http://www.w3.org/2001/XMLSchema#anyURI". Devuelve un "http://www.w3.org/2001/XMLSchema#boolean". El primer argumento es una expresión regular, y el segundo un URI. La función convierte el segundo argumento en tipo "http://www.w3.org/2001/XMLSchema#string" y posteriormente aplica "`urn:oasis:names:tc:xacml:1.0:function:string-regexp-match`".

```
urn:oasis:names:tc:xacml:2.0:function:ipAddress-regexp-match
```

Esta función determina si hay concordancia de una expresión regular. Tiene dos argumentos, el primero de los cuales es de tipo "http://www.w3.org/2001/XMLSchema#string" y el segundo de tipo "`urn:oasis:names:tc:xacml:2.0:data-type:ipAddress`". Devuelve un "http://www.w3.org/2001/XMLSchema#boolean". El primer argumento es una expresión regular y el segundo una dirección IPv4 o IPv6. La función convierte el segundo argumento en tipo "http://www.w3.org/2001/XMLSchema#string" y posteriormente aplica "`urn:oasis:names:tc:xacml:1.0:function:string-regexp-match`".

```
urn:oasis:names:tc:xacml:2.0:function:dnsName-regexp-match
```

Esta función determina si hay concordancia de una expresión regular. Tiene dos argumentos, el primero de los cuales es de tipo "http://www.w3.org/2001/XMLSchema#string" y el segundo de tipo "`urn:oasis:names:tc:xacml:2.0:data-type:dnsName`". Devuelve un "http://www.w3.org/2001/XMLSchema#boolean". El primer argumento es una expresión regular y el segundo un nombre DNS. La función convierte el segundo argumento en tipo "http://www.w3.org/2001/XMLSchema#string" y posteriormente aplica "`urn:oasis:names:tc:xacml:1.0:function:string-regexp-match`".

```
urn:oasis:names:tc:xacml:2.0:function:rfc822Name-regexp-match
```

Esta función determina si hay concordancia de una expresión regular. Tiene dos argumentos, el primero de los cuales es de tipo "http://www.w3.org/2001/XMLSchema#string" y el segundo de tipo "urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name". Devuelve un "http://www.w3.org/2001/XMLSchema#boolean". El primer argumento es una expresión regular y el segundo un nombre RFC 822. La función convierte el segundo argumento en tipo "http://www.w3.org/2001/XMLSchema#string" y posteriormente aplica "urn:oasis:names:tc:xacml:1.0:function:string-regexp-match".

urn:oasis:names:tc:xacml:2.0:function:x500Name-regexp-match

Esta función determina si hay concordancia de una expresión regular. Tiene dos argumentos, el primero de los cuales es de tipo "http://www.w3.org/2001/XMLSchema#string" y el segundo de tipo "urn:oasis:names:tc:xacml:1.0:data-type:x500Name". Devuelve un "http://www.w3.org/2001/XMLSchema#boolean". El primer argumento es una expresión regular y el segundo un nombre de directorio X.500. La función convierte el segundo argumento en tipo "http://www.w3.org/2001/XMLSchema#string" y posteriormente aplica "urn:oasis:names:tc:xacml:1.0:function:string-regexp-match".

A.3.14 Funciones de correspondencia especial

Estas funciones operan sobre varios tipos y el resultado de evaluación es "http://www.w3.org/2001/XMLSchema#boolean", aplicando el algoritmo de correspondencia estándar especificado.

urn:oasis:names:tc:xacml:1.0:function:x500Name-match

Esta función tiene dos argumentos de "urn:oasis:names:tc:xacml:2.0:data-type:x500Name" y devuelve un "http://www.w3.org/2001/XMLSchema#boolean". Devolverá "Verdadero" si y sólo si el primer argumento concuerda con una secuencia terminal de RDN del segundo argumento cuando se compara utilizando x500Name-equal.

urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match

Esta función tiene dos argumentos, el primero de los cuales es de tipo de datos "http://www.w3.org/2001/XMLSchema#string" y el segundo de tipo de datos "urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name". Devuelve un "http://www.w3.org/2001/XMLSchema#boolean". El resultado de la evaluación de esta función será "Verdadero" si el primer argumento concuerda con el segundo según W3C Datatypes:2001, 3.2.1.

Un nombre IETF RFC 822 consta de una parte local seguida de "@" y de una parte de dominio. La parte local es sensible a las mayúsculas y minúsculas, mientras que la parte de dominio (generalmente un nombre DNS) no lo es.

El segundo argumento contiene un rfc822Name completo. El primer argumento es un rfc822Name completo o parcial utilizado para seleccionar valores apropiados en el segundo argumento como se indica a continuación.

Para que haya concordancia con una determinada dirección del segundo argumento, el primer argumento debe especificar la dirección completa que es objeto de comparación. Para que haya concordancia con una dirección de un dominio concreto del segundo argumento, el primer argumento sólo tiene que especificar un nombre de dominio (generalmente un nombre DNS). Para que haya concordancia con una dirección de un dominio concreto del segundo argumento, el primer argumento debe especificar la parte del dominio deseada situando delante un ".".

A.3.15 Funciones basadas en XPath

En esta cláusula se especifican funciones que tienen expresiones XPath como argumentos. Como resultado de la evaluación de una expresión XPath se obtiene un conjunto de nodos, los nodos XML que corresponden a la expresión. Un nodo o un conjunto de nodos no aparecen en el sistema formal de tipo de datos de XACML. Todas las comparaciones y demás operaciones realizadas sobre los conjuntos de nodos se efectúan por separado de la función concreta especificada. Esto es, en esas funciones las expresiones XPath se restringen al contexto de petición XACML. El elemento <xacml-context:Request> es el nodo de contexto para cada expresión XPath. Se definen las siguientes funciones:

urn:oasis:names:tc:xacml:1.0:function:xpath-node-count

Esta función se aplica a un "http://www.w3.org/2001/XMLSchema#string" como argumento, que se interpreta como una expresión XPath, y de cuya evaluación se obtiene un "http://www.w3.org/2001/XMLSchema#integer". El valor que devuelve la función será el número de nodos del conjunto que concuerdan con la expresión XPath dada.

urn:oasis:names:tc:xacml:1.0:function:xpath-node-equal

Esta función se aplica a dos argumentos "http://www.w3.org/2001/XMLSchema#string", que se interpretan como expresiones XPath, y devuelve un "http://www.w3.org/2001/XMLSchema#boolean". La función devolverá "Verdadero" si alguno de los nodos XML del conjunto que concuerda con el primer argumento es igual a uno de los nodos XML del conjunto que concuerda con el segundo argumento. Dos nodos son iguales si tienen la misma identidad.

```
urn:oasis:names:tc:xacml:1.0:function:xpath-node-match
```

Esta función se aplica a dos argumentos "http://www.w3.org/2001/XMLSchema#string", que se interpretan como expresiones XPath y devuelve un "http://www.w3.org/2001/XMLSchema#boolean". El resultado de la evaluación de esta función será "Verdadero" si se cumple una de las dos condiciones siguientes:

1) alguno de los nodos XML del conjunto que concuerda con el primer argumento es igual a uno de los nodos XML del conjunto que concuerda con el segundo argumento;
2) hay un nodo de atributo o de elemento por debajo de alguno de los nodos XML del conjunto que concuerda con el primer argumento, que es igual a uno de los nodos XML del conjunto que concuerda con el segundo argumento.

Dos nodos son iguales si tienen la misma identidad.

NOTA – La primera condición es equivalente a "xpath-node-equal" y garantiza que "xpath-node-equal" es un caso especial de "xpath-node-match".

A.3.16 Tipos primitivos y funciones de ampliación

Las funciones y los tipos primitivos se determinan mediante identificadores de cadena que permiten introducir funciones distintas de las que ya especifica XACML. Este sistema permite ampliar el módulo XACML con funciones y tipos de datos primitivos especiales.

Con el fin de preservar la integridad de la estrategia de evaluación de XACML, el resultado de una función de ampliación dependerá sólo de los valores de sus argumentos. Los parámetros globales y ocultos no deben influir en la evaluación de una expresión. Las funciones no deben tener efectos secundarios, puesto que de otro modo no se podría garantizar una evaluación normalizada.

Anexo B

Identificadores XACML

En este anexo se definen identificadores normalizados para entidades de uso común.

B.1 Espacios de nombres XACML

En la actualidad hay dos espacios de nombres XACML definidos.

Las políticas se diseñan utilizando este identificador.

```
urn:oasis:names:tc:xacml:2.0:policy:schema:os
```

Los contextos de petición y respuesta se diseñan utilizando este identificador.

```
urn:oasis:names:tc:xacml:2.0:context:schema:os
```

B.2 Categorías de sujetos en el procedimiento de acceso

Este identificador indica la entidad del sistema que inició la petición de acceso o, en otras palabras, la entidad inicial de una cadena de petición. Si no se especifica la categoría del sujeto, es el valor por defecto.

```
urn:oasis:names:tc:xacml:1.0:subject-category:access-subject
```

Este identificador indica la entidad del sistema que recibirá los resultados de la petición (se utiliza cuando no es el sujeto iniciador (*access-subject*)).

```
urn:oasis:names:tc:xacml:1.0:subject-category:recipient-subject
```

Este identificador indica la entidad del sistema que trata la petición de acceso. Pueden ser varias. No se proporcionan medios para determinar el orden de tratamiento del mensaje.

```
urn:oasis:names:tc:xacml:1.0:subject-category:intermediary-subject
```

Este identificador indica una entidad del sistema asociada con una base de códigos local o remota que generó la petición. Los correspondientes atributos de sujeto pueden ser: el URL en el que se cargó la petición y/o la identidad del firmante del código. Puede haber varios. No se proporcionan medios para determinar el orden de tratamiento de la petición.

```
urn:oasis:names:tc:xacml:1.0:subject-category:codebase
```

Este identificador indica una entidad del sistema asociada con el ordenador que inició la petición de acceso. Un ejemplo de este tipo podría ser una identidad IPSEC.

```
urn:oasis:names:tc:xacml:1.0:subject-category:requesting-machine
```

B.3 Tipos de datos

Los siguientes identificadores indican los tipos de datos que se definen en A.2.

```
urn:oasis:names:tc:xacml:2.0:data-types:dayTimeDuration
urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration
urn:oasis:names:tc:xacml:1.0:data-type:x500Name.
urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name
urn:oasis:names:tc:xacml:2.0:data-type:ipAddress
urn:oasis:names:tc:xacml:2.0:data-type:dnsName
```

Los siguientes identificadores de tipo de dato se definen en W3C DataTypes2001.

```
http://www.w3.org/2001/XMLSchema#string
http://www.w3.org/2001/XMLSchema#boolean
http://www.w3.org/2001/XMLSchema#integer
http://www.w3.org/2001/XMLSchema#double
http://www.w3.org/2001/XMLSchema#time
http://www.w3.org/2001/XMLSchema#date
http://www.w3.org/2001/XMLSchema#dateTime
http://www.w3.org/2001/XMLSchema#anyURI
http://www.w3.org/2001/XMLSchema#hexBinary
http://www.w3.org/2001/XMLSchema#base64Binary
```

B.4 Atributos del sujeto

Estos identificadores indican atributos de un sujeto. Cuando se utilicen, será dentro de un elemento `<Subject>` del contexto de petición. Se accede a ellos por medio de un elemento `<SubjectAttributeDesignator>`, o de un elemento `<AttributeSelector>` que señale un elemento `<Subject>` del contexto de petición.

A lo sumo se asocia uno de estos atributos con cada sujeto. Cada atributo asociado con autenticación dentro de un elemento `<Subject>` se refiere al mismo evento de autenticación.

Este identificador indica el nombre del sujeto. El formato por defecto es "http://www.w3.org/2001/XMLSchema#string". Para indicar otros formatos se deben utilizar los atributos `DataType` de B.3.

```
urn:oasis:names:tc:xacml:1.0:subject:subject-id
```

Este identificador indica la categoría del sujeto. "access-subject" es el valor por defecto.

```
urn:oasis:names:tc:xacml:1.0:subject-category
```

Este identificador indica el dominio de seguridad del sujeto. Identifica al administrador y la política que gestiona el espacio de nombres en el que se administra el id del sujeto.

```
urn:oasis:names:tc:xacml:1.0:subject:subject-id-qualifier
```

Este identificador indica una clave pública utilizada para confirmar la identidad del sujeto.

```
urn:oasis:names:tc:xacml:1.0:subject:key-info
```

Este identificador indica el momento en el que se autenticó al sujeto.

```
urn:oasis:names:tc:xacml:1.0:subject:authentication-time
```

Este identificador indica el método utilizado para autenticar al sujeto.

```
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:authentication-
method
```

Este identificador indica el momento en el que el sujeto inició la petición de acceso, de acuerdo con el PEP.

```
urn:oasis:names:tc:xacml:1.0:subject:request-time
```

Este identificador indica el momento en el que comenzó la sesión en curso del sujeto, de acuerdo con el PEP.

```
urn:oasis:names:tc:xacml:1.0:subject:session-start-time
```

Los identificadores que aparecen a continuación indican la ubicación en la que se activaron las credenciales de autenticación. Tienen por objeto soportar las correspondientes entidades de la declaración de autenticación SAML.

Este identificador indica que la ubicación se expresa como dirección IP.

```
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:ip-address
```

El atributo correspondiente será del tipo de datos "http://www.w3.org/2001/XMLSchema#string".

Este identificador indica que la ubicación se expresa como nombre DNS.

```
urn:oasis:names:tc:xacml:1.0:subject:authn-locality:dns-name
```

El atributo correspondiente será del tipo de datos "http://www.w3.org/2001/XMLSchema#string".

Cuando ya se ha definido un atributo adecuado en LDAP, el identificador XACML se formará añadiendo el nombre del atributo al URI de IETF LDAP RFC (véase IETF RFC 2256). Por ejemplo, el nombre del atributo para la `contraseña de usuario (userPassword)` definida en IETF RFC 2256 será:

```
http://www.ietf.org/rfc/rfc2256.txt#userPassword
```

B.5 Atributos del recurso

Estos identificadores indican atributos del recurso. Los atributos correspondientes pueden aparecer en el elemento `<Resource>` del contexto de petición y se puede acceder a ellos por medio de un elemento `<ResourceAttributeDesignator>`, o de un elemento `<AttributeSelector>` que señale el elemento `<Resource>` del contexto de petición.

Este atributo identifica el recurso al que se solicita acceso. Si se proporciona un elemento `<xacml-context:ResourceContent>`, el recurso al que se quiere acceder será la totalidad del recurso, o una parte del mismo, proporcionado en el elemento `<xacml-context:ResourceContent>`.

```
urn:oasis:names:tc:xacml:1.0:resource:resource-id
```

Este atributo identifica el espacio de nombres del elemento de mayor rango del contenido del elemento `<xacml-context:ResourceContent>`. Si el contenido del recurso se proporciona en el contexto de petición y el espacio de nombre del recurso está definido en el recurso, el PDP confirmará si el espacio de nombre definido por ese atributo es el mismo que el definido en el recurso. El atributo correspondiente será del tipo "http://www.w3.org/2001/XMLSchema#anyURI".

```
urn:oasis:names:tc:xacml:2.0:resource:target-namespace
```

B.6 Atributos de la acción

Estos identificadores indican atributos de la acción que se solicita. Cuando se utilicen, será dentro del elemento `<Action>` del contexto de petición. Se accede a ellos por medio de un elemento `<ActionAttributeDesignator>`, o de un elemento `<AttributeSelector>` que señale el elemento `<Action>` del contexto de petición.

Este atributo identifica la acción para la que se solicita el acceso.

```
urn:oasis:names:tc:xacml:1.0:action:action-id
```

Cuando la acción está implícita, el valor del atributo `action-id` será:

```
urn:oasis:names:tc:xacml:1.0:action:implied-action
```

Este atributo identifica el espacio de nombre en el que se define el atributo `action-id`.

```
urn:oasis:names:tc:xacml:1.0:action:action-namespace
```

B.7 Atributos del entorno

Estos identificadores indican atributos del entorno dentro del cual se evalúa la petición de decisión. Cuando se utilicen en la petición de decisión, será dentro del elemento `<Environment>` del contexto de petición. Se accede a ellos por medio de un elemento `<EnvironmentAttributeDesignator>`, o de un elemento `<AttributeSelector>` que señale el elemento `<Environment>` del contexto de petición.

Este identificador indica el momento presente en el controlador del contexto. En la práctica, es el momento en el que se creó el contexto de petición. Por este motivo, si estos identificadores aparecen en múltiples ubicaciones dentro de un `<Policy>` o `<PolicySet>`, se asignará el mismo valor cada vez en el procedimiento de evaluación, con independencia del tiempo transcurrido entre dos casos.

```
urn:oasis:names:tc:xacml:1.0:environment:current-time
```

El atributo correspondiente será del tipo de datos "http://www.w3.org/2001/XMLSchema#time".

```
urn:oasis:names:tc:xacml:1.0:environment:current-date
```

El atributo correspondiente será del tipo de datos "http://www.w3.org/2001/XMLSchema#date".

```
urn:oasis:names:tc:xacml:1.0:environment:current-dateTime
```

El atributo correspondiente será del tipo de datos "http://www.w3.org/2001/XMLSchema#dateTime".

B.8 Códigos de estado

Se definen los códigos de estado que aparecen a continuación.

Este identificador indica que el proceso ha sido satisfactorio.

```
urn:oasis:names:tc:xacml:1.0:status:ok
```

Este identificador indica que no estaban disponibles todos los atributos necesarios para efectuar una decisión de política.

```
urn:oasis:names:tc:xacml:1.0:status:missing-attribute
```

Este identificador indica que algún valor de atributo contenía un error de sintaxis, por ejemplo, una letra en un campo numérico.

```
urn:oasis:names:tc:xacml:1.0:status:syntax-error
```

Este identificador indica que ocurrió un error durante la evaluación de la política, por ejemplo, una división por cero.

```
urn:oasis:names:tc:xacml:1.0:status:processing-error
```

B.9 Algoritmos de combinación

El algoritmo de combinación de reglas "deny-overrides" (denegación prioritaria) tiene el siguiente valor para el atributo ruleCombiningAlgId:

```
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides
```

El algoritmo de combinación de políticas "deny-overrides" (denegación prioritaria) tiene el siguiente valor para el atributo policyCombiningAlgId:

```
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:deny-overrides
```

El algoritmo de combinación de reglas "permit-overrides" (permiso prioritario) tiene el siguiente valor para el atributo ruleCombiningAlgId:

```
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides
```

El algoritmo de combinación de políticas "permit-overrides" (permiso-prioritario) tiene el siguiente valor para el atributo policyCombiningAlgId:

```
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides
```

El algoritmo de combinación de reglas "first-applicable" (se aplica el primero) tiene el siguiente valor para el atributo ruleCombiningAlgId:

```
urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable
```

El algoritmo de combinación de políticas "first-applicable" (se aplica el primero) tiene el siguiente valor para el atributo policyCombiningAlgId:

```
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:first-applicable
```

El algoritmo de combinación de políticas "only-one-applicable-policy" (sólo una política es aplicable) tiene el siguiente valor para el atributo policyCombiningAlgId:

```
urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:only-one-applicable
```

El algoritmo de combinación de reglas "ordered-deny-overrides" (denegación prioritaria por orden) tiene el siguiente valor para el atributo `ruleCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-deny-overrides
```

El algoritmo de combinación de políticas "ordered-deny-overrides" (denegación prioritaria por orden) tiene el siguiente valor para el atributo `policyCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.1:policy-combining-algorithm:ordered-deny-overrides
```

El algoritmo de combinación de reglas "ordered-permit-overrides" (permiso prioritario por orden) tiene el siguiente valor para el atributo `ruleCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.1:rule-combining-algorithm:ordered-permit-overrides
```

El algoritmo de combinación de políticas "ordered-permit-overrides" (permiso prioritario por orden) tiene el siguiente valor para el atributo `policyCombiningAlgId`:

```
urn:oasis:names:tc:xacml:1.1:policy-combining-algorithm:ordered-permit-overrides
```

Anexo C

Algoritmos de combinación

Este anexo contiene una descripción de los algoritmos de combinación de reglas y políticas especificados por XACML.

C.1 Deny-overrides (denegación prioritaria)

C.1.1 En esta cláusula se define el algoritmo de combinación de reglas "Deny-overrides" de una política.

Si el resultado de evaluación de una de las reglas de una política es "Denegar" (*Deny*), el resultado de la combinación de reglas será "Denegar". Si en evaluación de alguna regla se obtiene "Permitir" y en la de todas las demás reglas se obtiene "NotApplicable", el resultado de la combinación de reglas será "Permitir". En otras palabras, "Denegar" tiene prioridad, con independencia del resultado de la evaluación de cualquiera de las otras reglas de la combinación. Si todas las reglas devuelven "NotApplicable" a la petición de decisión, el resultado de la evaluación de la combinación de reglas será "NotApplicable".

Si ocurre un error cuando se evalúa el objetivo o la condición de una regla cuyo valor de efecto es "Denegar", la evaluación continuará con las siguientes reglas en busca de un resultado "Denegar". Si ninguna otra regla implica "Denegar", el resultado de la evaluación de la combinación será "Indeterminate", con el estado de error correspondiente.

Si hay al menos una regla que devuelva "Permitir" en la evaluación, si todas las demás reglas que no tienen errores de evaluación devuelven "Permitir" o "NotApplicable" y todas las reglas que sí tienen errores de evaluación contienen efectos "Permitir", el resultado de la combinación será "Permitir".

El siguiente seudocódigo representa la estrategia de evaluación de este algoritmo de combinación de reglas.

```
Decision denyOverridesRuleCombiningAlgorithm(Rule rule[])
{
   Boolean atLeastOneError  = false;
   Boolean potentialDeny    = false;
   Boolean atLeastOnePermit = false;
   for( i=0 ; i < lengthOf(rules) ; i++ )
   {
          Decision decision = evaluate(rule[i]);
          if (decision == Deny)
          {
                return Deny;
          }
          if (decision == Permit)
          {
                atLeastOnePermit = true;
                continue;
          }
          if (decision == NotApplicable)
          {
                continue;
          }
          if (decision == Indeterminate)
          {
                atLeastOneError = true;

                if (effect(rule[i]) == Deny)
                {
                      potentialDeny = true;
                }
                continue;
          }
   }
   if (potentialDeny)
   {
          return Indeterminate;
   }
   if (atLeastOnePermit)
   {
          return Permit;
   }
```

```
    if (atLeastOneError)
    {
           return Indeterminate;
    }
    return NotApplicable;
}
```

C.1.2 En esta cláusula se define el algoritmo de combinación de políticas "Deny-overrides" de un conjunto de políticas.

Si el resultado de evaluación de una de las políticas del conjunto es "Denegar" (*Deny*), el resultado de la combinación de políticas será "Denegar". En otras palabras, "Denegar" tiene prioridad, con independencia del resultado de la evaluación de cualquiera de las demás políticas del conjunto. Si todas las políticas devuelven "NotApplicable" a la petición de decisión, el resultado de la evaluación del conjunto de políticas será "NotApplicable".

Si ocurre un error cuando se evalúa el objetivo de una política, o se considera que una referencia a una política no es válida o si el resultado de la evaluación de una política es "Indeterminate", el resultado de la evaluación del conjunto de políticas será "Denegar".

El siguiente seudocódigo representa la estrategia de evaluación de este algoritmo de combinación de políticas.

```
Decision denyOverridesPolicyCombiningAlgorithm(Policy policy[])
{
    Boolean atLeastOnePermit = false;
    for( i=0 ; i < lengthOf(policy) ; i++ )
    {
           Decision decision = evaluate(policy[i]);
           if (decision == Deny)
           {
                  return Deny;
           }
           if (decision == Permit)
           {
                  atLeastOnePermit = true;
                  continue;
           }
           if (decision == NotApplicable)
           {
                  continue;
           }
           if (decision == Indeterminate)
           {
                  return Deny;
           }
    }
    if (atLeastOnePermit)
    {
           return Permit;
    }
    return NotApplicable;
}
```

Las obligaciones de cada política se combinan según se describe en 7.6.14.

C.2 Ordered-deny-overrides (denegación prioritaria por orden)

En el siguiente párrafo se define el algoritmo de combinación de reglas "Ordered-deny-overrides" de una política.

El proceso de este algoritmo es idéntico al de combinación de reglas Deny-overrides con una salvedad: las reglas deben evaluarse en el orden en que aparecen en la política.

En el siguiente párrafo se define el algoritmo de combinación de políticas "Ordered-deny-overrides" de un conjunto de políticas.

El proceso de este algoritmo es idéntico al de combinación de políticas Deny-overrides con una salvedad: las políticas deben evaluarse en el orden en que aparecen en el conjunto de políticas.

C.3 Permit-overrides (permiso-prioritario)

C.3.1 En esta cláusula se define el algoritmo de combinación de reglas "Permit-overrides" de una política.

Si el resultado de evaluación de una de las reglas de la política es "Permitir", el resultado de la combinación de reglas será "Permitir". Si hay alguna regla que implica "Denegar" y todas las demás reglas implican "NotApplicable", el resultado de la evaluación de la política será "Denegar". En otras palabras, "Permitir" tiene prioridad, con independencia del resultado de la evaluación de cualquiera de las demás reglas de la política. Si todas las reglas devuelven "NotApplicable" a la petición de decisión, el resultado de la evaluación de la política será "NotApplicable".

Si ocurre un error cuando se evalúa el objetivo o la condición de una regla cuyo efecto es "Permitir", la evaluación continuará buscando un resultado "Permitir". Si ninguna otra regla implica "Permitir", el resultado de la evaluación de la política será "Indeterminate", con el estado de error correspondiente.

Si hay al menos una regla que devuelva "Denegar" en la evaluación, si todas las demás reglas que no tienen errores de evaluación devuelven "Denegar" o "NotApplicable" y todas las reglas que sí tienen errores de evaluación contienen un valor de efecto "Denegar", el resultado de la evaluación de la política será "Denegar".

El siguiente seudocódigo representa la estrategia de evaluación de este algoritmo de combinación de reglas.

```
Decision permitOverridesRuleCombiningAlgorithm(Rule rule[])
{
   Boolean atLeastOneError  = false;
   Boolean potentialPermit  = false;
   Boolean atLeastOneDeny   = false;
   for( i=0 ; i < lengthOf(rule) ; i++ )
   {
          Decision decision = evaluate(rule[i]);
          if (decision == Deny)
          {
                atLeastOneDeny = true;
                continue;
          }
          if (decision == Permit)
          {
                return Permit;
          }
          if (decision == NotApplicable)
          {
                continue;
          }
          if (decision == Indeterminate)
          {
                atLeastOneError = true;

                if (effect(rule[i]) == Permit)
                {
                      potentialPermit = true;
                }
                continue;
          }
   }
   if (potentialPermit)
   {
          return Indeterminate;
   }
   if (atLeastOneDeny)
   {
          return Deny;
   }
   if (atLeastOneError)
   {
          return Indeterminate;
   }
   return NotApplicable;
}
```

C.3.2 En esta cláusula se define el algoritmo de combinación de políticas "Permit-overrides" de un conjunto de políticas.

Si el resultado de evaluación de una de las políticas del conjunto es "Permitir", el resultado de la combinación de políticas será "Permitir". En otras palabras, "Permitir" tiene prioridad, con independencia del resultado de la evaluación de cualquiera de las demás políticas del conjunto. Si todas las políticas devuelven "NotApplicable" a la petición de decisión, el resultado de la evaluación del conjunto de políticas será "NotApplicable".

Si ocurre un error cuando se evalúa el objetivo de una política, o se considera que no es válida una referencia a una política o el resultado de la evaluación de una política es "Indeterminate" el resultado de la evaluación del conjunto de políticas será "Indeterminate", con el estado de error correspondiente, siempre que no haya otras políticas que impliquen "Permitir" o "Denegar".

El siguiente seudocódigo representa la estrategia de evaluación de este algoritmo de combinación de políticas.

```
Decision permitOverridesPolicyCombiningAlgorithm(Policy policy[])
{
   Boolean atLeastOneError = false;
   Boolean atLeastOneDeny  = false;
   for( i=0 ; i < lengthOf(policy) ; i++ )
   {
          Decision decision = evaluate(policy[i]);
          if (decision == Deny)
          {
                atLeastOneDeny = true;
                continue;
          }
          if (decision == Permit)
          {
                return Permit;
          }
          if (decision == NotApplicable)
          {
                continue;
          }
          if (decision == Indeterminate)
          {
                atLeastOneError = true;
                continue;
          }
   }
   if (atLeastOneDeny)
   {
          return Deny;
   }
   if (atLeastOneError)
   {
          return Indeterminate;
   }
   return NotApplicable;
}
```

Las obligaciones de cada política se combinan según se describe en 7.6.14.

C.4 Ordered-permit-overrides (permiso prioritario por orden)

En el párrafo siguiente define el algoritmo de combinación de reglas "Ordered-permit-overrides" de una política.

El proceso de este algoritmo es idéntico al de combinación de reglas "Permit-overrides" con una salvedad: las reglas deben evaluarse en el orden en que aparecen en la política.

En el siguiente párrafo se define el algoritmo de combinación de políticas "Ordered-permit-overrides" de un conjunto de políticas.

El proceso de este algoritmo es idéntico al de combinación de políticas "Permit-overrides" con una salvedad: las políticas deben evaluarse en el orden en que aparecen en el conjunto de políticas.

C.5 First-applicable (se aplica el primero)

C.5.1 En esta cláusula se define el algoritmo de combinación de reglas "First-applicable" de una política.

Cada regla se evalúa en el orden que sigue en la política. Para una regla en particular, si hay concordancia del objetivo y el resultado de evaluación de la condición es "Verdadero", la evaluación de la política se interrumpirá y el resultado de la evaluación de la política será el correspondiente efecto de la regla (es decir, "Permitir" o "Denegar"). Para una regla en concreto seleccionada el resultado de evaluación, si el resultado de evaluación del objetivo o la condición es "Falso", se evaluará la regla siguiente. Si en dicho orden ya no queda por evaluar ninguna regla más, el resultado de la evaluación de la política será "NotApplicable".

Si ocurre un error cuando se evalúa el objetivo o la condición de una regla, la evaluación se interrumpirá, y el resultado de la evaluación de la política será "Indeterminate", con el estado de error correspondiente.

El siguiente seudocódigo representa la estrategia de evaluación de este algoritmo de combinación de reglas.

```
Decision firstApplicableEffectRuleCombiningAlgorithm(Rule rule[])
{
   for( i = 0 ; i < lengthOf(rule) ; i++ )
   {
          Decision decision = evaluate(rule[i]);
          if (decision == Deny)
          {
                return Deny;
          }
          if (decision == Permit)
          {
                return Permit;
          }
          if (decision == NotApplicable)
          {
                continue;
          }
          if (decision == Indeterminate)
          {
                return Indeterminate;
          }
   }
   return NotApplicable;
}
```

C.5.2 En esta cláusula se define el algoritmo de combinación de políticas "First-applicable" de un conjunto de políticas.

Cada política se evalúa en el orden en que aparece en el conjunto de políticas. Para una política en concreto, si el resultado de evaluación del objetivo es "Verdadero" y el resultado de la evaluación de la política es un valor determinado "Permitir" o "Denegar", la evaluación se interrumpirá y el resultado de la evaluación del conjunto de políticas será el valor de efecto de esa política. Para una política en concreto, si el resultado de la evaluación del objetivo es "Falso" o el resultado de la evaluación de la política es "NotApplicable", se evaluará la política siguiente. Si en dicho orden ya no queda por evaluar ninguna política más, el resultado de la evaluación del conjunto de políticas será "NotApplicable".

Si ocurre un error cuando se evalúa el objetivo, o una política específica, si se considera que no es valida la referencia a la política, o el resultado de la evaluación de la política es "Indeterminate", la evaluación del algoritmo de combinación de políticas se interrumpirá, y el resultado de la evaluación del conjunto de políticas será "Indeterminate", con el estado de error correspondiente.

El siguiente seudocódigo representa la estrategia de evaluación de este algoritmo de combinación de políticas.

```
Decision firstApplicableEffectPolicyCombiningAlgorithm(Policy policy[])
{
    for( i = 0 ; i < lengthOf(policy) ; i++ )
    {
        Decision decision = evaluate(policy[i]);
        if(decision == Deny)
        {
            return Deny;
        }
        if(decision == Permit)
        {
            return Permit;
        }
        if (decision == NotApplicable)
        {
            continue;
        }
        if (decision == Indeterminate)
        {
            return Indeterminate;
        }
    }
    return NotApplicable;
}
```

Las obligaciones de cada política se combinan según se describe en 7.6.14.

C.6 Only-one-applicable (sólo una es aplicable)

En esta cláusula se define el algoritmo de combinación de políticas "Only-one-applicable" de un conjunto de políticas.

En el conjunto completo de políticas del conjunto, si se considera que ninguna política es aplicable en virtud de su objetivo, el resultado del algoritmo de combinación de políticas será "NotApplicable". Si se considera que hay más de una política que resulta aplicable en virtud de su objetivo, el resultado del algoritmo de combinación de políticas será "Indeterminate".

Si al evaluar el objetivo se considera que sólo una política es aplicable, el resultado del algoritmo de combinación de políticas será el resultado de evaluar esa política.

Si ocurre un error cuando se evalúa el objetivo de una política, o se considera que no es válida una referencia a una política, o el resultado de la evaluación de la política es "Indeterminate", el resultado de la evaluación del conjunto de políticas será "Indeterminate", con el estado de error correspondiente.

El siguiente seudocódigo representa la estrategia de evaluación de este algoritmo de combinación de políticas.

```
Decision onlyOneApplicablePolicyPolicyCombiningAlogrithm(Policy policy[])
{
  Boolean          atLeastOne     = false;
  Policy           selectedPolicy = null;
  ApplicableResult appResult;

  for ( i = 0; i < lengthOf(policy) ; i++ )
  {
     appResult = isApplicable(policy[I]);

     if ( appResult == Indeterminate )
     {
         return Indeterminate;
     }
     if( appResult == Applicable )
     {
         if ( atLeastOne )
         {
             return Indeterminate;
         }
         else
         {
             atLeastOne     = true;
             selectedPolicy = policy[i];
         }
     }
     if ( appResult == NotApplicable )
     {
         continue;
     }
  }
  if ( atLeastOne )
  {
      return evaluate(selectedPolicy);
  }
  else
  {
      return NotApplicable;
  }
}
```

Anexo D

Esquema XACML

D.1 Esquema de contexto XACML

Esta cláusula proporciona el esquema de contexto XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <xs:import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
      schemaLocation="http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-
schema-os.xsd"/>
      <!--   -->
      <xs:element name="Request" type="xacml-context:RequestType"/>
      <xs:complexType name="RequestType">
            <xs:sequence>
                  <xs:element ref="xacml-context:Subject" maxOccurs="unbounded"/>
                  <xs:element ref="xacml-context:Resource" maxOccurs="unbounded"/>
                  <xs:element ref="xacml-context:Action"/>
                  <xs:element ref="xacml-context:Environment"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="Response" type="xacml-context:ResponseType"/>
      <xs:complexType name="ResponseType">
            <xs:sequence>
                  <xs:element ref="xacml-context:Result" maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="Subject" type="xacml-context:SubjectType"/>
      <xs:complexType name="SubjectType">
            <xs:sequence>
                  <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="SubjectCategory" type="xs:anyURI"
            default="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="Resource" type="xacml-context:ResourceType"/>
      <xs:complexType name="ResourceType">
            <xs:sequence>
                  <xs:element ref="xacml-context:ResourceContent" minOccurs="0"/>
                  <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="ResourceContent" type="xacml-context:ResourceContentType"/>
      <xs:complexType name="ResourceContentType" mixed="true">
            <xs:sequence>
                  <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="Action" type="xacml-context:ActionType"/>
      <xs:complexType name="ActionType">
            <xs:sequence>
                  <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
```

```
      </xs:complexType>
      <!-- -->
      <xs:element name="Environment" type="xacml-context:EnvironmentType"/>
      <xs:complexType name="EnvironmentType">
            <xs:sequence>
                  <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="Attribute" type="xacml-context:AttributeType"/>
      <xs:complexType name="AttributeType">
            <xs:sequence>
                  <xs:element ref="xacml-context:AttributeValue"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
            <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
            <xs:attribute name="Issuer" type="xs:string" use="optional"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="AttributeValue" type="xacml-context:AttributeValueType"/>
      <xs:complexType name="AttributeValueType" mixed="true">
            <xs:sequence>
                  <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute namespace="##any" processContents="lax"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="Result" type="xacml-context:ResultType"/>
      <xs:complexType name="ResultType">
            <xs:sequence>
                  <xs:element ref="xacml-context:Decision"/>
                  <xs:element ref="xacml-context:Status" minOccurs="0"/>
                  <xs:element ref="xacml:Obligations" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="ResourceId" type="xs:string" use="optional"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="Decision" type="xacml-context:DecisionType"/>
      <xs:simpleType name="DecisionType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="Permit"/>
                  <xs:enumeration value="Deny"/>
                  <xs:enumeration value="Indeterminate"/>
                  <xs:enumeration value="NotApplicable"/>
            </xs:restriction>
      </xs:simpleType>
      <!-- -->
      <xs:element name="Status" type="xacml-context:StatusType"/>
      <xs:complexType name="StatusType">
            <xs:sequence>
                  <xs:element ref="xacml-context:StatusCode"/>
                  <xs:element ref="xacml-context:StatusMessage" minOccurs="0"/>
                  <xs:element ref="xacml-context:StatusDetail" minOccurs="0"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="StatusCode" type="xacml-context:StatusCodeType"/>
      <xs:complexType name="StatusCodeType">
            <xs:sequence>
                  <xs:element ref="xacml-context:StatusCode" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="Value" type="xs:anyURI" use="required"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="StatusMessage" type="xs:string"/>
      <!-- -->
      <xs:element name="StatusDetail" type="xacml-context:StatusDetailType"/>
```

```
      <xs:complexType name="StatusDetailType">
            <xs:sequence>
                  <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="MissingAttributeDetail" type="xacml-
context:MissingAttributeDetailType"/>
      <xs:complexType name="MissingAttributeDetailType">
            <xs:sequence>
                  <xs:element ref="xacml-context:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
            <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
            <xs:attribute name="Issuer" type="xs:string" use="optional"/>
      </xs:complexType>
      <!-- -->
</xs:schema>
```

D.2 Esquema de políticas

Esta cláusula proporciona el esquema de políticas XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <!-- -->
      <xs:element name="PolicySet" type="xacml:PolicySetType"/>
      <xs:complexType name="PolicySetType">
            <xs:sequence>
                  <xs:element ref="xacml:Description" minOccurs="0"/>
                  <xs:element ref="xacml:PolicySetDefaults" minOccurs="0"/>
                  <xs:element ref="xacml:Target"/>
                  <xs:choice minOccurs="0" maxOccurs="unbounded">
                        <xs:element ref="xacml:PolicySet"/>
                        <xs:element ref="xacml:Policy"/>
                        <xs:element ref="xacml:PolicySetIdReference"/>
                        <xs:element ref="xacml:PolicyIdReference"/>
                        <xs:element ref="xacml:CombinerParameters"/>
                        <xs:element ref="xacml:PolicyCombinerParameters"/>
                        <xs:element ref="xacml:PolicySetCombinerParameters"/>
                  </xs:choice>
                  <xs:element ref="xacml:Obligations" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="PolicySetId" type="xs:anyURI" use="required"/>
            <xs:attribute name="Version" type="xacml:VersionType" default="1.0"/>
            <xs:attribute name="PolicyCombiningAlgId" type="xs:anyURI" use="required"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="CombinerParameters" type="xacml:CombinerParametersType"/>
      <xs:complexType name="CombinerParametersType">
            <xs:sequence>
                  <xs:element ref="xacml:CombinerParameter" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="CombinerParameter" type="xacml:CombinerParameterType"/>
      <xs:complexType name="CombinerParameterType">
            <xs:sequence>
                  <xs:element ref="xacml:AttributeValue"/>
            </xs:sequence>
            <xs:attribute name="ParameterName" type="xs:string" use="required"/>
      </xs:complexType>
      <!-- -->
```

```
      <xs:element name="RuleCombinerParameters"
type="xacml:RuleCombinerParametersType"/>
      <xs:complexType name="RuleCombinerParametersType">
            <xs:complexContent>
                  <xs:extension base="xacml:CombinerParametersType">
                        <xs:attribute name="RuleIdRef" type="xs:string"
use="required"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="PolicyCombinerParameters"
type="xacml:PolicyCombinerParametersType"/>
      <xs:complexType name="PolicyCombinerParametersType">
            <xs:complexContent>
                  <xs:extension base="xacml:CombinerParametersType">
                        <xs:attribute name="PolicyIdRef" type="xs:anyURI"
use="required"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="PolicySetCombinerParameters"
type="xacml:PolicySetCombinerParametersType"/>
      <xs:complexType name="PolicySetCombinerParametersType">
            <xs:complexContent>
                  <xs:extension base="xacml:CombinerParametersType">
                        <xs:attribute name="PolicySetIdRef" type="xs:anyURI"
use="required"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="PolicySetIdReference" type="xacml:IdReferenceType"/>
      <xs:element name="PolicyIdReference" type="xacml:IdReferenceType"/>
      <!-- -->
      <xs:element name="PolicySetDefaults" type="xacml:DefaultsType"/>
      <xs:element name="PolicyDefaults" type="xacml:DefaultsType"/>
      <xs:complexType name="DefaultsType">
            <xs:sequence>
                  <xs:choice>
                        <xs:element ref="xacml:XPathVersion"/>
                  </xs:choice>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="XPathVersion" type="xs:anyURI"/>
      <!-- -->
      <xs:complexType name="IdReferenceType">
            <xs:simpleContent>
                  <xs:extension base="xs:anyURI">
                        <xs:attribute name="Version" type="xacml:VersionMatchType"
use="optional"/>
                        <xs:attribute name="EarliestVersion"
type="xacml:VersionMatchType" use="optional"/>
                        <xs:attribute name="LatestVersion"
type="xacml:VersionMatchType" use="optional"/>
                  </xs:extension>
            </xs:simpleContent>
      </xs:complexType>
      <!-- -->
      <xs:simpleType name="VersionType">
            <xs:restriction base="xs:string">
                  <xs:pattern value="(\d+\.)*\d+"/>
            </xs:restriction>
      </xs:simpleType>
      <!-- -->
      <xs:simpleType name="VersionMatchType">
            <xs:restriction base="xs:string">
                  <xs:pattern value="((\d+|\*)\.)*(\d+|\*|\+)"/>
            </xs:restriction>
```

```
    </xs:simpleType>
    <!-- -->
    <xs:element name="Policy" type="xacml:PolicyType"/>
    <xs:complexType name="PolicyType">
          <xs:sequence>
                <xs:element ref="xacml:Description" minOccurs="0"/>
                <xs:element ref="xacml:PolicyDefaults" minOccurs="0"/>
                <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
                <xs:element ref="xacml:Target"/>
                <xs:choice maxOccurs="unbounded">
                      <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
                      <xs:element ref="xacml:RuleCombinerParameters" minOccurs="0"/>
                      <xs:element ref="xacml:VariableDefinition"/>
                      <xs:element ref="xacml:Rule"/>
                </xs:choice>
                <xs:element ref="xacml:Obligations" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="PolicyId" type="xs:anyURI" use="required"/>
          <xs:attribute name="Version" type="xacml:VersionType" default="1.0"/>
          <xs:attribute name="RuleCombiningAlgId" type="xs:anyURI" use="required"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="Description" type="xs:string"/>
    <!-- -->
    <xs:element name="Rule" type="xacml:RuleType"/>
    <xs:complexType name="RuleType">
          <xs:sequence>
                <xs:element ref="xacml:Description" minOccurs="0"/>
                <xs:element ref="xacml:Target" minOccurs="0"/>
                <xs:element ref="xacml:Condition" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="RuleId" type="xs:string" use="required"/>
          <xs:attribute name="Effect" type="xacml:EffectType" use="required"/>
    </xs:complexType>
    <!-- -->
    <xs:simpleType name="EffectType">
          <xs:restriction base="xs:string">
                <xs:enumeration value="Permit"/>
                <xs:enumeration value="Deny"/>
          </xs:restriction>
    </xs:simpleType>
    <!-- -->
    <xs:element name="Target" type="xacml:TargetType"/>
    <xs:complexType name="TargetType">
          <xs:sequence>
                <xs:element ref="xacml:Subjects" minOccurs="0"/>
                <xs:element ref="xacml:Resources" minOccurs="0"/>
                <xs:element ref="xacml:Actions" minOccurs="0"/>
                <xs:element ref="xacml:Environments" minOccurs="0"/>
          </xs:sequence>
    </xs:complexType>
    <!-- -->
    <xs:element name="Subjects" type="xacml:SubjectsType"/>
    <xs:complexType name="SubjectsType">
          <xs:sequence>
                <xs:element ref="xacml:Subject" maxOccurs="unbounded"/>
          </xs:sequence>
    </xs:complexType>
    <!-- -->
    <xs:element name="Subject" type="xacml:SubjectType"/>
    <xs:complexType name="SubjectType">
          <xs:sequence>
                <xs:element ref="xacml:SubjectMatch" maxOccurs="unbounded"/>
          </xs:sequence>
    </xs:complexType>
    <!-- -->
    <xs:element name="Resources" type="xacml:ResourcesType"/>
    <xs:complexType name="ResourcesType">
          <xs:sequence>
                <xs:element ref="xacml:Resource" maxOccurs="unbounded"/>
          </xs:sequence>
```

```
	</xs:complexType>
	<!-- -->
	<xs:element name="Resource" type="xacml:ResourceType"/>
	<xs:complexType name="ResourceType">
		<xs:sequence>
			<xs:element ref="xacml:ResourceMatch" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!-- -->
	<xs:element name="Actions" type="xacml:ActionsType"/>
	<xs:complexType name="ActionsType">
		<xs:sequence>
			<xs:element ref="xacml:Action" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!-- -->
	<xs:element name="Action" type="xacml:ActionType"/>
	<xs:complexType name="ActionType">
		<xs:sequence>
			<xs:element ref="xacml:ActionMatch" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!-- -->
	<xs:element name="Environments" type="xacml:EnvironmentsType"/>
	<xs:complexType name="EnvironmentsType">
		<xs:sequence>
			<xs:element ref="xacml:Environment" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!-- -->
	<xs:element name="Environment" type="xacml:EnvironmentType"/>
	<xs:complexType name="EnvironmentType">
		<xs:sequence>
			<xs:element ref="xacml:EnvironmentMatch" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<!-- -->
	<xs:element name="SubjectMatch" type="xacml:SubjectMatchType"/>
	<xs:complexType name="SubjectMatchType">
		<xs:sequence>
			<xs:element ref="xacml:AttributeValue"/>
			<xs:choice>
				<xs:element ref="xacml:SubjectAttributeDesignator"/>
				<xs:element ref="xacml:AttributeSelector"/>
			</xs:choice>
		</xs:sequence>
		<xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
	</xs:complexType>
	<!-- -->
	<xs:element name="ResourceMatch" type="xacml:ResourceMatchType"/>
	<xs:complexType name="ResourceMatchType">
		<xs:sequence>
			<xs:element ref="xacml:AttributeValue"/>
			<xs:choice>
				<xs:element ref="xacml:ResourceAttributeDesignator"/>
				<xs:element ref="xacml:AttributeSelector"/>
			</xs:choice>
		</xs:sequence>
		<xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
	</xs:complexType>
	<!-- -->
	<xs:element name="ActionMatch" type="xacml:ActionMatchType"/>
	<xs:complexType name="ActionMatchType">
		<xs:sequence>
			<xs:element ref="xacml:AttributeValue"/>
			<xs:choice>
				<xs:element ref="xacml:ActionAttributeDesignator"/>
				<xs:element ref="xacml:AttributeSelector"/>
			</xs:choice>
		</xs:sequence>
		<xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
```

```
      </xs:complexType>
      <!-- -->
      <xs:element name="EnvironmentMatch" type="xacml:EnvironmentMatchType"/>
      <xs:complexType name="EnvironmentMatchType">
            <xs:sequence>
                  <xs:element ref="xacml:AttributeValue"/>
                  <xs:choice>
                        <xs:element ref="xacml:EnvironmentAttributeDesignator"/>
                        <xs:element ref="xacml:AttributeSelector"/>
                  </xs:choice>
            </xs:sequence>
            <xs:attribute name="MatchId" type="xs:anyURI" use="required"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="VariableDefinition" type="xacml:VariableDefinitionType"/>
      <xs:complexType name="VariableDefinitionType">
            <xs:sequence>
                  <xs:element ref="xacml:Expression"/>
            </xs:sequence>
            <xs:attribute name="VariableId" type="xs:string" use="required"/>
      </xs:complexType>
      <!-- -->
      <xs:element name="Expression" type="xacml:ExpressionType" abstract="true"/>
      <xs:complexType name="ExpressionType" abstract="true"/>
      <!-- -->
      <xs:element name="VariableReference"
      type="xacml:VariableReferenceType" substitutionGroup="xacml:Expression"/>
      <xs:complexType name="VariableReferenceType">
            <xs:complexContent>
                  <xs:extension base="xacml:ExpressionType">
                        <xs:attribute name="VariableId" type="xs:string"
use="required"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="AttributeSelector"
      type="xacml:AttributeSelectorType" substitutionGroup="xacml:Expression"/>
      <xs:complexType name="AttributeSelectorType">
            <xs:complexContent>
                  <xs:extension base="xacml:ExpressionType">
                        <xs:attribute name="RequestContextPath" type="xs:string"
use="required"/>
                        <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                        <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="ResourceAttributeDesignator"
      type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
      <xs:element name="ActionAttributeDesignator"
      type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
      <xs:element name="EnvironmentAttributeDesignator"
      type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
      <!-- -->
      <xs:complexType name="AttributeDesignatorType">
            <xs:complexContent>
                  <xs:extension base="xacml:ExpressionType">
                        <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
                        <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                        <xs:attribute name="Issuer" type="xs:string" use="optional"/>
                        <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
                  </xs:extension>
            </xs:complexContent>
```

```
      </xs:complexType>
      <!-- -->
      <xs:element name="SubjectAttributeDesignator"
      type="xacml:SubjectAttributeDesignatorType" substitutionGroup="xacml:Expression"/>
      <xs:complexType name="SubjectAttributeDesignatorType">
            <xs:complexContent>
                  <xs:extension base="xacml:AttributeDesignatorType">
                        <xs:attribute name="SubjectCategory" type="xs:anyURI"
                        use="optional" default="urn:oasis:names:tc:xacml:1.0:subject-
category:access-subject"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="AttributeValue" type="xacml:AttributeValueType"
            substitutionGroup="xacml:Expression"/>
      <xs:complexType name="AttributeValueType" mixed="true">
            <xs:complexContent mixed="true">
                  <xs:extension base="xacml:ExpressionType">
                        <xs:sequence>
                              <xs:any namespace="##any" processContents="lax"
minOccurs="0"
                                          maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                        <xs:anyAttribute namespace="##any" processContents="lax"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="Function" type="xacml:FunctionType"
substitutionGroup="xacml:Expression"/>
      <xs:complexType name="FunctionType">
            <xs:complexContent>
                  <xs:extension base="xacml:ExpressionType">
                        <xs:attribute name="FunctionId" type="xs:anyURI"
use="required"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="Condition" type="xacml:ConditionType"/>
      <xs:complexType name="ConditionType">
            <xs:sequence>
                  <xs:element ref="xacml:Expression"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
      <xs:element name="Apply" type="xacml:ApplyType"
substitutionGroup="xacml:Expression"/>
      <xs:complexType name="ApplyType">
            <xs:complexContent>
                  <xs:extension base="xacml:ExpressionType">
                        <xs:sequence>
                              <xs:element ref="xacml:Expression" minOccurs="0"
maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="FunctionId" type="xs:anyURI"
use="required"/>
                  </xs:extension>
            </xs:complexContent>
      </xs:complexType>
      <!-- -->
      <xs:element name="Obligations" type="xacml:ObligationsType"/>
      <xs:complexType name="ObligationsType">
            <xs:sequence>
                  <xs:element ref="xacml:Obligation" maxOccurs="unbounded"/>
            </xs:sequence>
      </xs:complexType>
      <!-- -->
```

```
     <xs:element name="Obligation" type="xacml:ObligationType"/>
     <xs:complexType name="ObligationType">
            <xs:sequence>
                  <xs:element ref="xacml:AttributeAssignment" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="ObligationId" type="xs:anyURI" use="required"/>
            <xs:attribute name="FulfillOn" type="xacml:EffectType" use="required"/>
     </xs:complexType>
     <!-- -->
     <xs:element name="AttributeAssignment" type="xacml:AttributeAssignmentType"/>
     <xs:complexType name="AttributeAssignmentType" mixed="true">
            <xs:complexContent mixed="true">
                  <xs:extension base="xacml:AttributeValueType">
                        <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
                  </xs:extension>
            </xs:complexContent>
     </xs:complexType>
     <!-- -->
</xs:schema>
```

D.3 Esquema de protocolo SAML en XACML

Esta cláusula proporciona el esquema de protocolo en SAML XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:xacml:2.0:saml:protocol:schema:os"
    xmlns:xs="http://www.23.org/2001/XMLSchema"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
    xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
  <xs:import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
      schemaLocation="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=security"/>
  <xs:import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
      schemaLocation="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=security"/>
  <xs:import namespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
      schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-
context-schema-os.xsd"/>
  <xs:import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
      schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-
policy-schema-os.xsd"/>
  <xs:annotation>
    <xs:documentation>
        Document identifier: access_control-xacml-2.0-saml-protocol-schema-os.xsd
        Location: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-
protocol-schema-os.xsd
    </xs:documentation>
  </xs:annotation>
  <!--    -->
  <xs:element name="XACMLAuthzDecisionQuery"
           type="XACMLAuthzDecisionQueryType"/>
  <xs:complexType name="XACMLAuthzDecisionQueryType">
    <xs:complexContent>
      <xs:extension base="samlp:RequestAbstractType">
        <xs:sequence>
          <xs:element ref="xacml-context:Request"/>
        </xs:sequence>
```

```
        <xs:attribute name="InputContextOnly"
                      type="boolean"
                      use="optional"
                      default="false"/>
        <xs:attribute name="ReturnContext"
                      type="boolean"
                      use="optional"
                      default="false"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!--    -->
  <xs:element name="XACMLPolicyQuery"
           type="XACMLPolicyQueryType"/>
  <xs:complexType name="XACMLPolicyQueryType">
    <xs:complexContent>
      <xs:extension base="samlp:RequestAbstractType">
        <xs:choice minOccurs="0" maxOccurs="unbounded">>
          <xs:element ref="xacml-context:Request"/>
          <xs:element ref="xacml:Target"/>
          <xs:element ref="xacml:PolicySetIdReference"/>
          <xs:element ref="xacml:PolicyIdReference"/>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</schema>
```

D.4 Esquema de aserción SAML en XACML

Esta cláusula proporciona el esquema de aserción SAML en XACML.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:xacml:2.0:saml:assertion:schema:os"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:xs="http://www.23.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
    xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
  <xs:import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
      schemaLocation="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=security"/>
  <xs:import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
      schemaLocation="http://www.oasis-
open.org/committees/tc_home.php?wg_abbrev=security"/>
  <xs:import namespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
      schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-
context-schema-os.xsd"/>
  <xs:import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
      schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-
policy-schema-os.xsd"/>
  <xs:annotation>
    <xs:documentation>
      Document identifier: access_control-xacml-2.0-saml-assertion-schema-cd-02.xsd
      Location: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-
assertion-schema-cd-os.xsd
    </xs:documentation>
  </xs:annotation>
  <!--    -->
  <xs:element name="XACMLAuthzDecisionStatement"
           type="XACMLAuthzDecisionStatementType"/>
```

```
  <xs:complexType name="XACMLAuthzDecisionStatementType">
    <xs:complexContent>
      <xs:extension base="samlp:StatementAbstractType">
        <xs:sequence>
          <xs:element ref="xacml-context:Response"/>
          <xs:element ref="xacml-context:Request"  MinOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!--    -->
  <xs:element name="XACMLPolicyStatement"
           type="XACMLPolicyStatementType"/>
  <xs:complexType name="XACMLPolicyStatementType">
    <xs:complexContent>
      <xs:extension base="samlp:StatementAbstractType">
        <xs:choice minOccurs="0" maxOccurs="unbounded">>
          <xs:element ref="xacml:Policy"/>
          <xs:element ref="xacml:PolicySet"/>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</schema>
```

Apéndice I

Consideraciones sobre la seguridad

En este apéndice se identifican algunos casos en los que puede haber dudas sobre la seguridad y la privacidad, que se deben considerar al implementar un sistema basado en XACML. Son tareas del implementador decidir si estas situaciones de peligro pueden producirse efectivamente en su entorno y seleccionar las protecciones apropiadas.

I.1 Modelo de amenaza

Se supone que el adversario tiene acceso al canal de comunicación utilizado por las partes del diálogo XACML, y que es capaz de interpretar, insertar, borrar y modificar los mensajes o parte de los mismos.

Además, una parte puede utilizar información obtenida en un antiguo mensaje con fines dolosos en futuras transacciones. Por otro lado, se asume que las reglas y políticas son de fiar sólo en la misma medida que lo son las partes que las crean y emplean. Los mecanismos para el establecimiento de la confianza están fuera del alcance de esta Recomendación.

Los mensajes transmitidos entre partes en el modelo XACML pueden ser atacados por terceros malintencionados. Otros puntos vulnerables son el PEP, el PDP y el PAP. Si bien algunas de estas entidades no forman parte en realidad del alcance de esta Recomendación, una duda de seguridad en ellas puede suponer la duda de seguridad del control de acceso que realiza el PEP.

Puede haber un peligro en otros componentes de un sistema distribuido, por ejemplo un sistema operativo o el sistema de nombres de dominio (DNS), que están fuera del alcance de este análisis de modelos de amenazas. Un peligro en estos componentes puede llevar a su vez a una violación de la política.

En las siguientes cláusulas se dan ejemplos de peligros específicos que pueden ser pertinentes a un sistema XACML.

I.1.1 Revelación no autorizada

Como el sistema XACML no establece ningún mecanismo inherente para proteger la confidencialidad de los mensajes intercambiados entre las partes, un adversario podría observar los mensajes en tránsito. Bajo ciertas políticas de seguridad, la revelación de esta información es una violación. La revelación de los atributos o los tipos de las peticiones de decisión que un sujeto presente puede ser una violación de la política de privacidad. En el sector comercial, las consecuencias de la revelación no autorizada de los datos personales pueden variar: es una imprudencia bochornosa para el custodio, pero también motivo de encarcelamiento y grandes multas si se trata de la revelación de datos médicos o financieros.

La protección de la confidencialidad previene la revelación no autorizada.

I.1.2 Reproducción de mensajes

Un ataque por reproducción de mensajes se produce cuando el adversario graba y reproduce mensajes legítimos entre las partes del diálogo XACML. Este ataque puede provocar la denegación del servicio, la utilización de información obsoleta o la simulación.

La prevención de ataques de reproducción requiere la utilización de protecciones basadas en la validez de los mensajes.

La criptación del mensaje no impide un ataque de reproducción, puesto que simplemente se reproduce, sin que el adversario lo comprenda.

I.1.3 Inserción de mensajes

Un ataque por inserción de mensajes se produce cuando el adversario inserta mensajes en la secuencia de mensajes que se transmiten las partes del diálogo XACML.

La solución a dichos ataques es el empleo de protecciones de autenticación mutua y de integridad de la secuencia de los mensajes entre las partes. La autenticación mutua SSL no es una protección suficiente, sólo demuestra que la otra parte es la identificada por el sujeto del certificado X.509. A fin de ser efectivos, es necesario confirmar que el sujeto del certificado está autorizado para enviar los mensajes.

I.1.4 Supresión de mensajes

Un ataque por supresión de mensajes se produce cuando el adversario borra mensajes en la secuencia de mensajes enviados entre las partes del diálogo XACML. La supresión de mensajes puede conducir a la denegación del servicio; sin embargo, un sistema XACML diseñado apropiadamente no debería dar una decisión de autorización incorrecta por efecto de un ataque por supresión de mensajes.

La solución a este tipo de ataque es la utilización de protecciones de integridad de la secuencia de mensajes transmitidos entre las partes.

I.1.5 Modificación del mensaje

Si un adversario puede interceptar un mensaje y cambiar su contenido, entonces se podría alterar una decisión de autorización. Una salvaguarda de la integridad del mensaje puede evitar un ataque para modificar el mismo.

I.1.6 Resultados NotApplicable

Si el resultado es "NotApplicable", el PDP no ha podido localizar una política con un objetivo que corresponda a la información de la petición de decisión. En general, se recomienda aplicar una política con efecto "Denegar", de modo que en lugar de que un PDP devuelva "NotApplicable", devolvería "Denegar".

Sin embargo, en algunos modelos de seguridad, como los que se encuentran en muchos servidores web, una decisión de autorización "NotApplicable" se trata como equivalente a "Permitir". Hay consideraciones de seguridad especiales a tener en cuenta para la seguridad de este procedimiento, explicadas en los párrafos siguientes.

Si "NotApplicable" debe tratarse como "Permitir", resulta de vital importancia que los algoritmos de correspondencia utilizados por la política para contrastar los elementos de la petición de decisión correspondan exactamente a la sintaxis de datos utilizada por las aplicaciones que enviarán la petición de decisión. Si no hay concordancia, el resultado será "NotApplicable", siendo tratado como "Permitir". Por tanto, un fallo de concordancia no intencionado puede permitir el acceso no intencionado.

Los respondedores http comerciales permiten tratar varias sintaxis de manera equivalente. Puede usarse "%" para representar caracteres en hexadecimal. La dirección URL "/../" facilita muchos modos de especificar el mismo valor, pudiendo permitirse varios conjuntos de caracteres, y en algunos casos, el mismo carácter impreso puede representarse mediante diferentes valores binarios. A menos que el algoritmo de correspondencia utilizado por la política sea lo suficientemente sofisticado para detectar estas variaciones, podría permitirse un acceso no intencionado.

Puede ser seguro tratar "NotApplicable" como "Permitir" únicamente en un entorno cerrado, en el que pueda garantizarse que todas las aplicaciones que formulan una petición de decisión utilicen la sintaxis exacta que esperan las políticas. En un entorno más abierto, en el que puede haber peticiones de decisión de aplicaciones que utilizan cualquier sintaxis válida, se recomienda encarecidamente que no se trate "NotApplicable" como "Permitir" a menos que las reglas de correspondencia estén muy bien estudiadas para que haya concordancia con todas las entradas aplicables posibles, independientemente de las variaciones de sintaxis o de tipo. Un PEP debe rechazar el acceso a menos que reciba una decisión de autorización "Permitir" explícita.

I.1.7 Reglas negativas

Una regla negativa es la que se basa en un predicado distinto de "True". Si no se utilizan con cuidado, las reglas negativas pueden provocar una violación de política. Por tanto, algunas autoridades no recomiendan su uso. Sin embargo, como las reglas negativas pueden ser extremadamente eficientes en ciertos casos, se ha decidido incluirlas en XACML, pero se recomienda que se utilicen con atención, evitándolas a ser posible.

Una utilización frecuente de las reglas negativas es rechazar el acceso a un individuo o subgrupo, aunque su pertenencia a un grupo mayor pudiera permitirles tener acceso. Por ejemplo, se podría querer escribir una regla que permita a todos los vicepresidentes ver los datos financieros no publicados, excepto a Joe, que es solamente un vicepresidente honorífico y que podría ser indiscreto. Si se tiene un control total sobre la administración de los atributos de sujeto, la mejor solución sería definir "Vicepresidente" y "Vicepresidente honorífico" como grupos distintos, y luego definir las reglas en función de ello. No obstante, en determinados entornos no será viable esta solución (merece la pena constatar que, generalmente, referirse a individuos en reglas no es fácilmente extensible, prefiriéndose atributos compartidos).

Si no se utilizan con cuidado, las reglas negativas pueden provocar una violación de política en dos casos frecuentes: al omitir los atributos y al cambiar el grupo de base. Un ejemplo de atributos omitidos sería una política de autorización de acceso, a menos que el sujeto sea menos fiable. Una intervención que ocultara el atributo de menor fiabilidad al PDP podría resultar en un acceso no autorizado. En algunos entornos, el sujeto podría suprimir la publicación de atributos por parte de la aplicación de controles de privacidad, o el servidor o depósito que alberga la información podría no estar disponible debido a motivos accidentales o intencionados.

Un ejemplo de cambio del grupo de base sería una política en la que todas las personas del departamento de ingeniería pudieran cambiar el código fuente informático, excepto las secretarias. Supóngase que el departamento fusiona con otro departamento de ingeniería. Se quiere mantener la misma política, pero el nuevo departamento incluye también individuos identificados como asistentes administrativos, que tienen los mismos privilegios que las secretarias. A menos que se altere la política, se les permitirá de manera no intencionada cambiar el código fuente informático. Es fácil evitar este tipo de problemas cuando un individuo administra todas las políticas, pero cuando se distribuye la administración, como permite el XACML, hay que instalar protecciones explícitas contra este tipo de situaciones.

I.2 Salvaguardas

I.2.1 Autenticación

La autenticación permite a una parte en una transacción determinar la identidad de la otra parte, pudiendo ser en una dirección, o bidireccional.

Dado el carácter sensible de los sistemas de control de acceso, es importante que un PEP autentique la identidad del PDP al que envía las peticiones de decisión. De lo contrario, existe el riesgo de que un adversario pueda facilitar decisiones de autorización falsas o no válidas, produciendo una violación de política.

Es igual de importante que un PDP autentique la identidad del PEP y valore el nivel de confianza para determinar, si los hubiera, qué datos sensibles pueden pasarse. Téngase presente que incluso las simples respuestas "Permitir" o "Denegar" podrían explotarse si un adversario pudiera hacer peticiones ilimitadas a un PDP.

Pueden utilizarse muchas técnicas diferentes para la autenticación: código colocalizado, red privada, VPN o firmas digitales. La autenticación puede llevarse a cabo también como parte del protocolo de comunicación utilizado para intercambiar los contextos. En este caso, la autenticación puede llevarse a cabo a nivel de mensaje o de sesión.

I.2.2 Administración de políticas

Si el contenido de las políticas se expone fuera del sistema de control de acceso, algunos sujetos podrían utilizar esta información para determinar cómo obtener acceso no autorizado.

Para evitar esta amenaza puede ser necesario un control de acceso en el mismo depósito utilizado para almacenar las políticas. Además, el elemento `<Status>` debería utilizarse para devolver valores de atributos que faltan únicamente cuando la exposición de las identidades de dichos atributos no comprometa la seguridad.

I.2.3 Confidencialidad

Los mecanismos de confidencialidad aseguran que sólo los destinatarios podrán leer el contenido de un mensaje, y no cualquiera que vea el mensaje mientras que éste transita. Existen dos áreas en las que debe considerarse la confidencialidad: durante la transmisión y dentro de un elemento `<Policy>`.

I.2.3.1 Confidencialidad de comunicación

En algunos entornos se considera una buena práctica tratar como confidenciales todos los datos dentro de un sistema de control de acceso. En otros entornos, las políticas podrían ponerse libremente a disposición para ser distribuidas, inspeccionadas y auditadas. Si la información de políticas se mantiene secreta es para dificultar que un adversario sepa los pasos que serían suficientes para obtener un acceso no autorizado. Independientemente del enfoque escogido, la seguridad del sistema de control de acceso no debería depender del secreto de la política.

Cualquier consideración sobre seguridad en relación a la transmisión o intercambio de elementos XACML `<Policy>` queda fuera del ámbito de la norma XACML. Muchas veces es importante asegurar la integridad y la confidencialidad de los elementos `<Policy>` intercambiados entre dos partes, pero los implementadores deberán determinar los mecanismos adecuados para su propio entorno.

Un mecanismo de confidencialidad, como SSL, puede facilitar confidencialidad en las comunicaciones. La utilización de un esquema de punto a punto como SSL puede presentar otras vulnerabilidades si hay peligro en uno de los puntos extremos.

I.2.3.2 Confidencialidad de enunciados

En algunos casos es conveniente que la implementación sólo encripte partes de un elemento `<Policy>` XACML.

Puede utilizarse la norma Encryption:2002 de W3C para encriptar la totalidad o partes de un documento XML, recomendándose su uso con XACML.

Ni que decir tiene que si se utiliza un depósito para facilitar la comunicación de una política de texto claro (es decir, no encriptado) entre el PAP y el PDP, debería utilizarse un depósito seguro para almacenar esos datos sensibles.

I.2.4 Integridad de política

La política XACML, utilizada por el PDP para evaluar el contexto de petición, es el corazón del sistema. Por tanto, resulta esencial mantener su integridad, que tiene dos aspectos. El primero es asegurar que los elementos `<Policy>` no se han alterado desde que se crearon inicialmente por el PAP. El segundo es asegurar que no se han insertado o borrado elementos `<Policy>` del conjunto de políticas.

En muchos casos se pueden dar las dos garantías garantizando la integridad de las partes e implementando mecanismos de sesión para asegurar la comunicación entre ellas. La selección de los mecanismos adecuados se deja a los implementadores. Sin embargo, cuando se distribuye la política entre organizaciones para que la apliquen posteriormente, o cuando la política viaja con el recurso protegido, sería útil firmar dicha política. En tales casos, se recomienda utilizar con XACML la norma de sintaxis y procesado de la firma XML del W3C.

Las firmas digitales sólo deberían utilizarse para asegurar la integridad de los enunciados, y no como método de selección o evaluación de políticas. Es decir, que el PDP no debería solicitar una política utilizando como criterios la identidad del firmante, o en si ha sido firmada o no (puesto que tal base de selección sería en sí una política). Sin embargo, el PDP debe verificar si la clave utilizada para firmar la política está controlada por el pretendido expedidor de la política. El medio de hacer esto depende de la tecnología de firma específica escogida, quedando fuera del alcance de esta Recomendación.

I.2.5 Identificadores de políticas

Puesto que los identificadores pueden referenciar políticas, el PAP tiene que definir identificadores únicos. La confusión entre identificadores podría provocar la identificación errónea de la política aplicable. Esta Recomendación no se pronuncia sobre si un PAP debe generar un nuevo identificador cuando se modifica una política, o puede utilizar el mismo identificador en la política modificada. Es una cuestión de práctica administrativa, pero se debe prestar atención en ambos casos. Si se reutiliza el identificador, existe el peligro de que otras políticas o conjuntos de políticas que lo referencian se ven afectadas negativamente. Si se utiliza un nuevo identificador, estas otras políticas pueden seguir utilizando la política anterior, a menos que se elimine. En cualquier caso, los resultados podrían no ser los pretendidos por el administrador de políticas.

I.2.6 Modelo de confianza

Las discusiones sobre autenticación, integridad y salvaguarda de confidencialidad suponen necesariamente un modelo de confianza subyacente: ¿cómo puede deducir una de las partes que una clave dada corresponde únicamente a una determinada parte, identificada, de manera que la clave pueda utilizarse para criptar datos para esa parte o verificar sus firmas (u otras estructuras de integridad)? Existen muchos tipos diferentes de modelos de confianza: jerarquías estrictas, autoridades distribuidas, la Web, el puente, etc.

Merece la pena considerar las relaciones entre las diversas partes del sistema de control de acceso desde el punto de vista de las interdependencias o de la no interdependencia.

- Ninguna de las entidades del sistema de autorización depende del PEP. Pueden recopilar datos de él, por ejemplo datos de autenticación, pero son responsables de verificarlos ellas mismas.
- El correcto funcionamiento del sistema depende de la habilidad del PEP para imponer realmente decisiones de políticas.
- El PEP depende del PDP para evaluar políticas correctamente. Ello implica que, a su vez, se proporcionen las entradas correctas al PDP. Aparte de esto, el PDP no depende del PEP.
- El PDP depende del PAP para proporcionar políticas adecuadas. El PAP no depende de otros componentes.

I.2.7 Privacidad

Es importante ser consciente de que cualquier transacción que tenga lugar en relación al control de acceso puede revelar información privada sobre las partes. Por ejemplo, si una política XACML afirma que ciertos datos sólo pueden ser leídos por sujetos que tienen una calidad "superior", entonces cualquier transacción en la que se permite a un sujeto tener acceso a esos datos desvela información a un adversario sobre el estado del sujeto. Las consideraciones sobre privacidad pueden crear requisitos de criptación y/o control de acceso para lo referente a la propia aplicación de políticas XACML: canales protegidos de manera confidencial para mensajes de protocolo de petición/respuesta, protección de atributos de sujeto almacenados y en tránsito, etc.

La selección y el uso de mecanismos de privacidad adecuados para un determinado entorno quedan fuera del alcance de XACML. La decisión sobre cómo y cuándo se utilizan tales mecanismos, o si se utilizan, se deja a los implementadores asociados al entorno.

Apéndice II

Ejemplos de XACML

Este apéndice contiene dos ejemplos ilustrativos, del uso de XACML. El primero, relativamente simple, tiene por objeto ilustrar el uso de los atributos del objetivo, el contexto, las funciones de correspondencia y el sujeto. El segundo ejemplo ilustra además el uso del algoritmo de combinación de reglas, las condiciones y las obligaciones.

II.1 Primer ejemplo

Esta cláusula contiene el primer ejemplo.

II.1.1 Ejemplo de política

Supongamos que una sociedad llamada Medi Corp (identificada mediante su nombre de dominio: med.example.com), en inglés, tiene la siguiente política de control de acceso que:

Se permite a todo usuario que tenga un nombre de correo electrónico en el espacio de nombres "med.example.com" que realice cualquier acción sobre cualquier recurso.

Una política XACML consta de información de encabezamiento, una descripción explícita facultativa de la política, un **objetivo**, una o más **reglas** y un conjunto de **obligaciones** facultativo.

```
[a02]   <?xml version="1.0" encoding="UTF-8"?>
[a03]   <Policy
[a04]   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
[a05]   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
[a06]   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
[a07]   PolicyId="urn:oasis:names:tc:example:SimplePolicy1"
[a08]   RuleCombiningAlgId="identifier:rule-combining-algorithm:deny-overrides">
[a09]    <Description>
[a10]     Medi Corp access control policy
[a11]    </Description>
[a12]    <Target/>
[a13]    <Rule
[a14]    RuleId= "urn:oasis:names:tc:xacml:2.0:example:SimpleRule1"
[a15]    Effect="Permit">
[a16]     <Description>
[a17]      Any subject with an e-mail name in the med.example.com domain
[a18]      can perform any action on any resource.
[a19]     </Description>
[a20]     <Target>
[a21]      <Subjects>
[a22]       <Subject>
[a23]        <SubjectMatch
[a24]        MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
[a25]         <AttributeValue
[a26]         DataType="http://www.w3.org/2001/XMLSchema#string">
[a27]          med.example.com
[a28]         </AttributeValue>
[a29]         <SubjectAttributeDesignator
[a30]         AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
[a31]         DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
[a32]        </SubjectMatch>
[a33]       </Subject>
[a34]      </Subjects>
[a35]     </Target>
[a36]    </Rule>
[a37]   </Policy>
```

[a02] Es una etiqueta estándar del documento XML que indica cuál es la versión de XML que se está utilizando y el tipo de codificación de caracteres.

[a03] Introduce la política XACML propiamente dicha.

[a04] – [a05] Son declaraciones del espacio de nombres XML.

[a06] Proporciona un URN para el esquema de políticas XACML.

[a07] Asigna un nombre a este ejemplar de política. El nombre de una política debe ser único para un PDP dado a fin de que no exista ambigüedad si se hace referencia a una política desde otra. Se omite el atributo de versión, por lo que se adopta el valor por defecto "1.0".

[a08] Especifica el algoritmo que se utilizará para determinar los resultados de las distintas reglas que pueda contener la política. El algoritmo de combinación de reglas "deny-overrides" (denegación prioritaria) que se especifica indica que, si en la evaluación de alguna regla se obtiene "Denegar", la política debe devolver "Denegar". Si el resultado de la evaluación de todas las reglas es "Permitir", la política debe devolver "Permitir". El algoritmo de combinación de reglas, que se describe exhaustivamente en el anexo C, también determina el curso de acción cuando ocurre algún error al evaluar las reglas, y qué hacer con las reglas que no son aplicables a una petición de decisión particular.

[a09] – [a11] Proporcionan una descripción explícita de la política. Esta descripción es facultativa.

[a12] Describe las peticiones de decisión a las que se aplica esta política. Si el sujeto, el recurso, la acción y el entorno de una petición de decisión no concuerdan con los valores especificados en el objetivo de la política, no será necesario evaluar el resto de la política. Esta sección del objetivo es útil para crear un índice de un conjunto de políticas. En este ejemplo simple, la sección del objetivo indica que la política es aplicable a cualquier petición de decisión.

[a13] Introduce la única regla de esta política simple.

[a14] Especifica el identificador de esta regla. Al igual que las políticas, cada regla debe tener un identificador único (al menos que sea único para cualquier PDP que vaya a utilizar la política).

[a15] Señala el efecto que produce la regla si el resultado de la evaluación es "Verdadero". El efecto de una regla puede ser "Permitir" o "Denegar". En este caso, si se satisface la regla, devolverá "Permitir" en la evaluación, lo que significa que, en lo que respecta a esta regla, se debería permitir el acceso solicitado. Si en la evaluación de la regla se obtiene "Falso", devuelve un resultado de "NotApplicable". Si ocurre un error al evaluar la regla, el resultado será "Indeterminate". Como ya se ha señalado, el algoritmo de combinación de reglas para la política especifica cómo se combinan los diversos valores de las reglas para dar un solo valor de política.

[a16] – [a19] Proporcionan una descripción explícita de esta regla. Esta descripción es facultativa.

[a20] Introduce el objetivo de la regla. Como el objetivo de una política, el objetivo de una regla describe las peticiones de decisión a las que se aplica la regla. Si el sujeto, el recurso, la acción y el entorno de una petición de decisión no concuerdan con los valores especificados en el objetivo de la regla, no es necesario evaluar el resto de la regla, y el resultado de evaluación de la regla será "NotApplicable".

El objetivo de la regla es similar al de la propia política, con una diferencia importante: en [a23] – [a32] se especifica un valor con el que debe concordar el sujeto en la petición de decisión. El elemento `<SubjectMatch>` especifica una función de correspondencia en el atributo `MatchId`, un valor literal de "med.example.com" y un indicador de un atributo concreto del sujeto en el contexto de petición por medio del elemento `<SubjectAttributeDesignator>`. La función de correspondencia se utilizará para comparar el valor literal con el valor del atributo del sujeto. Esta regla sólo se aplicará a una petición de decisión particular si la evaluación de concordancia es "Verdadero". Si fuera "Falso", esta regla devolverá un valor "NotApplicable".

[a36] Cierra la regla. Esta regla está enteramente en el elemento `<Target>`. En reglas más complejas `<Target>` puede ir seguido de un elemento `<Condition>` (que también podría ser un conjunto de condiciones de tipo conjuntivo o disyuntivo al mismo tiempo).

[a37] Cierra la política. Como ya se ha explicado, esta política sólo tiene una regla, pero existen políticas más complejas que pueden tener cualquier número de reglas.

II.1.2 Ejemplo de contexto de petición

Considérese una petición de decisión hipotética que se podría presentar a un PDP que ejecuta la política anterior. En inglés, la petición de acceso que genera la petición de decisión se podría formular como sigue:

Bart Simpson, cuyo nombre de correo electrónico es "bs@simpsons.com" quiere leer su expediente médico en Medi Corp.

En XACML, la información de la petición de decisión se transcribe en la siguiente declaración de contexto de petición:

```
[a38]   <?xml version="1.0" encoding="UTF-8"?>
[a39]   <Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
[a40]   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
http://docs.oasis-
open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
[a41]    <Subject>
[a42]     <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
[a43]      <AttributeValue>
[a44]       bs@simpsons.com
[a45]      </AttributeValue>
[a46]     </Attribute>
[a47]    </Subject>
[a48]    <Resource>
[a49]     <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
[a50]      <AttributeValue>
[a51]       file://example/med/record/patient/BartSimpson
[a52]      </AttributeValue>
[a53]     </Attribute>
[a54]    </Resource>
[a55]    <Action>
[a56]     <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
[a57]      <AttributeValue>
[a58]       read
[a59]      </AttributeValue>
[a60]     </Attribute>
[a61]    </Action>
[a62]    <Environment/>
[a63]   </Request>
```

[a38] – [a40] Contienen la información de encabezamiento del contexto de petición, y se utilizan como el encabezamiento de la política explicada anteriormente.

El elemento `<Subject>` contiene uno o más atributos de la entidad que realiza la petición de acceso. Puede haber múltiples sujetos y cada uno de ellos puede tener distintos atributos. En el caso que nos ocupa, en [a41] – [a47] sólo hay un sujeto, y el sujeto tiene sólo un atributo: la identidad del sujeto, expresada en forma de nombre de correo electrónico, "bs@simpsons.com". En este ejemplo se omite el atributo de categoría del sujeto y, por lo tanto, se considera el valor por defecto "access-subject" (sujeto de acceso).

El elemento `<Resource>` contiene uno o más atributos del recurso para el que el sujeto (o sujetos) ha solicitado el acceso. Sólo puede haber un `<Resource>` por cada petición de decisión. Las líneas [a48] – [a54] contienen el único atributo del recurso para el que Bart Simpson ha solicitado acceso: el recurso identificado mediante el URI del fichero, esto es "**file**://`medico/record/patient/BartSimpson`".

El elemento `<Action>` contiene uno o más atributos de la acción que el sujeto (o sujetos) desea realizar sobre el recurso. Sólo puede haber una acción por cada petición de decisión. [a55] – [a61] describen la identidad de la acción que Bart Simpson desea realizar, es decir "read" (leer).

El elemento `<Environment>`, [a62], está vacío.

[a63] Cierra el contexto de petición. Un contexto de petición más complejo podría tener otros atributos no asociados con el sujeto, el recurso o la acción. Éstos se habrían ubicado en un elemento `<Environment>` facultativo detrás del elemento `<Action>`.

El PDP que procesa este contexto de petición sitúa la política en su depósito de políticas. Compara el sujeto, el recurso, la acción y el entorno del contexto de petición con los sujetos, recursos, acciones y entornos del objetivo de la política. Dado que el objetivo de la política está vacío, la política concuerda con este contexto.

El PDP compara a continuación el sujeto, el recurso, la acción y el entorno del contexto de petición con el objetivo de la única regla de esta política. El recurso solicitado concuerda con el elemento `<Target>` y la acción solicitada concuerda con el elemento `<Target>`, pero el atributo "`subject-id`" solicitante no coincide con "med.example.com".

II.1.3 Ejemplo de contexto de respuesta

Como resultado de la evaluación de la política, no hay ninguna regla de esta política que devuelva "Permitir" para esta petición. El algoritmo de combinación de reglas para la política especifica que, en este caso, el resultado será "NotApplicable". El código del contexto de respuesta es:

```
[a64]  <?xml version="1.0" encoding="UTF-8"?>
[a65]  <Response xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
http://docs.oasis-open.org/xacml/xacml-core-2.0-context-schema-os.xsd">
[a66]   <Result>
[a67]    <Decision>NotApplicable</Decision>
[a68]   </Result>
[a69]  </Response>
```

[a64] – [a65] Contienen la información de cabecera para la respuesta, como la información de la política comentada anteriormente.

El elemento `<Result>` de las líneas [a66] – [a68] contiene el resultado de evaluar la petición de decisión contrastada con la política. En el caso que nos ocupa, el resultado es "NotApplicable". Una política puede devolver "Permitir", "Denegar", "NotApplicable" o "Indeterminate". Por lo tanto, el PEP tendrá que denegar el acceso.

[a69] Cierra el contexto de respuesta.

II.2 Segundo ejemplo

Esta cláusula contiene un ejemplo de documento XML, un ejemplo de contexto de petición y un ejemplo de reglas XACML. El documento XML es un registro médico. Se definen cuatro reglas distintas que ilustran un algoritmo de combinación de reglas, así como condiciones y obligaciones.

II.2.1 Ejemplo de ejemplar de registro médico

A continuación se muestra un ejemplar de registro médico a la que se pueden aplicar las reglas XACML del ejemplo. El esquema `<record>` está definido en el espacio de nombres registrado que administra Medi Corp.

```
[a70]   <?xml version="1.0" encoding="UTF-8"?>
[a71]    <record xmlns="urn:example:med:schemas:record"
[a72]    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
[a73]     <patient>
[a74]      <patientName>
[a75]       <first>Bartholomew</first>
[a76]       <last>Simpson</last>
[a77]      </patientName>
[a78]      <patientContact>
[a79]       <street>27 Shelbyville Road</street>
[a80]       <city>Springfield</city>
[a81]       <state>MA</state>
[a82]       <zip>12345</zip>
[a83]       <phone>555.123.4567</phone>
[a84]       <fax/>
[a85]       <email/>
[a86]      </patientContact>
[a87]      <patientDoB>1992-03-21</patientDoB>
[a88]      <patientGender>male</patientGender>
[a89]      <patient-number>555555</patient-number>
[a90]     </patient>
[a91]     <parentGuardian>
[a92]      <parentGuardianId>HS001</parentGuardianId>
[a93]      <parentGuardianName>
[a94]       <first>Homer</first>
[a95]       <last>Simpson</last>
[a96]      </parentGuardianName>
[a97]      <parentGuardianContact>
[a98]       <street>27 Shelbyville Road</street>
[a99]       <city>Springfield</city>
[a100]      <state>MA</state>
[a101]      <zip>12345</zip>
[a102]      <phone>555.123.4567</phone>
[a103]      <fax/>
```

```
[a104]      <email>homers@aol.com</email>
[a105]     </parentGuardianContact>
[a106]    </parentGuardian>
[a107]    <primaryCarePhysician>
[a108]     <physicianName>
[a109]      <first>Julius</first>
[a110]      <last>Hibbert</last>
[a111]     </physicianName>
[a112]     <physicianContact>
[a113]      <street>1 First St</street>
[a114]      <city>Springfield</city>
[a115]      <state>MA</state>
[a116]      <zip>12345</zip>
[a117]      <phone>555.123.9012</phone>
[a118]      <fax>555.123.9013</fax>
[a119]      <email/>
[a120]     </physicianContact>
[a121]      <registrationID>ABC123</registrationID>
[a122]     </primaryCarePhysician>
[a123]     <insurer>
[a124]      <name>Blue Cross</name>
[a125]      <street>1234 Main St</street>
[a126]      <city>Springfield</city>
[a127]      <state>MA</state>
[a128]      <zip>12345</zip>
[a129]      <phone>555.123.5678</phone>
[a130]      <fax>555.123.5679</fax>
[a131]      <email/>
[a132]     </insurer>
[a133]    <medical>
[a134]     <treatment>
[a135]      <drug>
[a136]       <name>methylphenidate hydrochloride</name>
[a137]       <dailyDosage>30mgs</dailyDosage>
[a138]       <startDate>1999-01-12</startDate>
[a139]      </drug>
[a140]      <comment>
[a141]       patient exhibits side-effects of skin coloration and carpal
degeneration
[a142]      </comment>
[a143]     </treatment>
[a144]     <result>
[a145]      <test>blood pressure</test>
[a146]      <value>120/80</value>
[a147]      <date>2001-06-09</date>
[a148]      <performedBy>Nurse Betty</performedBy>
[a149]     </result>
[a150]    </medical>
[a151]   </record>
```

II.2.2 Ejemplo de contexto de petición

A continuación figura un ejemplo ilustrativo de un contexto de petición para el que se pueden aplicar las reglas del ejemplo. Representa una petición por parte del médico Julius Hibbert para leer la fecha de nacimiento del paciente en el registro de Bartholomew Simpson.

```
[a152]  <?xml version="1.0" encoding="UTF-8"?>
[a153]  <Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=" urn:oasis:names:tc:xacml:2.0:context:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-
os.xsd">
[a154]   <Subject>
[a155]    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject-
category"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
[a156]     <AttributeValue>urn:oasis:names:tc:xacml:1.0:subject-
category:access-subject</AttributeValue>
[a157]    </Attribute>
[a158]    <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
```

```
 DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="med.example.com">
[a159]     <AttributeValue>CN=Julius Hibbert</AttributeValue>
[a160]    </Attribute>
[a161]    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:name-
format"
DataType="http://www.w3.org/2001/XMLSchema#anyURI" Issuer="med.example.com">
[a162]     <AttributeValue>
[a163]      urn:oasis:names:tc:xacml:1.0:datatype:x500name
[a164]     </AttributeValue>
[a165]    </Attribute>
[a166]    <Attribute
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:role"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="med.example.com">
[a167]     <AttributeValue>physician</AttributeValue>
[a168]    </Attribute>
[a169]    <Attribute
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:physician-id"
DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="med.example.com">
[a170]     <AttributeValue>jh1234</AttributeValue>
[a171]    </Attribute>
[a172]   </Subject>
[a173]   <Resource>
[a174]    <ResourceContent>
[a175]     <md:record xmlns:md="urn:example:med:schemas:record"
xsi:schemaLocation="urn:example:med:schemas:record
http:www.med.example.com/schemas/record.xsd">
[a176]      <md:patient>
[a177]       <md:patientDoB>1992-03-21</md:patientDoB>
[a178]       <md:patient-number>555555</md:patient-number>
[a179]      </md:patient>
[a180]     </md:record>
[a181]    </ResourceContent>
[a182]    <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
[a183]     <AttributeValue>
[a184]      //med.example.com/records/bart-simpson.xml#
[a185]  xmlns(md=:Resource/ResourceContent/xpointer
[a186]  (/md:record/md:patient/md:patientDoB)
[a187]     </AttributeValue>
[a188]    </Attribute>
[a189]  </Resource>
[a190]   <Action>
[a191]    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-
id"
DataType="http://www.w3.org/2001/XMLSchema#string">
[a192]     <AttributeValue>read</AttributeValue>
[a193]    </Attribute>
[a194]   </Action>
[a195]   <Environment/>
[a196]  </Request>
```

[a152] – [a153] Declaraciones estándar del espacio de nombres.

[a154] – [a172] Los atributos del sujeto están en el elemento `<Subject>` del elemento `<Request>`. Cada atributo consta de metadatos de atributo y del valor del atributo. Sólo hay un sujeto implicado en esta petición.

[a155] – [a157] Cada elemento `<Subject>` tiene un atributo `SubjectCategory`. El valor de este atributo describe la función que desempeña el sujeto correspondiente en la realización de la petición de decisión. El valor "`access-subject`" indica la identidad para la que se expidió la petición.

[a158] – [a160] Atributo `subject-id` del sujeto.

[a161] – [a165] El formato de la identidad del sujeto.

[a166] – [a168] Atributo `role` del sujeto.

[a169] – [a171] Atributo `physician-id` del sujeto.

[a173] – [a189] Los atributos del recurso están en el elemento `<Resource>` del elemento `<Request>`. Cada atributo consta de metadatos de atributo y del valor del atributo.

[a174] – [a181] Contenido del recurso. Descripción del ejemplar del recurso XML para el que se puede pedir acceso completo o parcial.

[a182] – [a188] El identificador del ejemplar del recurso para el que se solicita el acceso, que es una expresión XPath dentro del elemento `<ResourceContent>` que selecciona los datos a los que se va a acceder.

[a190] – [a194] Los atributos de la acción están en el elemento `<Action>` del elemento `<Request>`.

[a192] Identificador de la acción.

[a195] El elemento `<Environment>` vacío.

II.2.3 Ejemplo de reglas en lenguaje explícito

Se deben respetar las siguientes reglas en lenguaje explícito:

1) Regla 1: Una persona, identificada mediante su número de paciente, puede leer cualquier registro en el que conste como paciente designado.
2) Regla 2: Una persona puede leer cualquier registro en el que conste como progenitor o tutor designado, cuando el paciente es menor de 16 años.
3) Regla 3: Un médico puede modificar cualquier elemento médico si está designado como médico de atención primaria, siempre que se avise al paciente por correo electrónico.
4) Regla 4: Los administradores no pueden leer ni modificar ningún elemento médico de un registro del paciente.

Estas reglas pueden escribirlas distintos PAP que funcionen de forma independiente, o un único PAP.

II.2.4 Ejemplo de ejemplares de reglas XACML

II.2.4.1 Regla 1

La regla 1 ilustra una regla simple con un único elemento `<Condition>`. También muestra el uso del elemento `<VariableDefinition>` para definir una función que se pueda utilizar para toda la política.

El siguiente ejemplar `<Rule>` XACML expresa la regla 1:

```
[a197]  <?xml version="1.0" encoding="UTF-8"?>
[a198]  <Policy
[a199]  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
[a200]  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=" urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-os.xsd"
[a201]  xmlns:md="http://www.med.example.com/schemas/record.xsd"
[a202]  PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:1"
[a203]  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
[a204]   <PolicyDefaults>
[a205]      <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-
19991116</XPathVersion>
[a206]   </PolicyDefaults>
[a207]   <Target/>
[a208]   <VariableDefinition VariableId="17590034">
[a209]    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a210]     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-
only">
[a211]      <SubjectAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:patient-number"
[a212]      DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a213]     </Apply>
[a214]     <Apply
[a215]     FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
[a216]      <AttributeSelector
[a217]      RequestContextPath="//xacml-context:Resource/xacml-
context:ResourceContent/md:record/md:patient/md:patient-number/text()"
[a218]      DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a219]     </Apply>
[a220]    </Apply>
[a221]   </VariableDefinition>
[a222]   <Rule
```

```
[a223]   RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:1"
[a224]   Effect="Permit">
[a225]    <Description>
[a226]      A person may read any medical record in the
[a227]      http://www.med.example.com/schemas/record.xsd namespace
[a228]      for which he or she is the designated patient
[a229]    </Description>
[a230]    <Target>
[a231]     <Resources>
[a232]      <Resource>
[a233]       <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
[a234]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a235]          urn:example:med:schemas:record
[a236]        </AttributeValue>
[a237]        <ResourceAttributeDesignator AttributeId=
[a238]        "urn:oasis:names:tc:xacml:2.0:resource:target-namespace"
[a239]        DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a240]       </ResourceMatch>
[a241]       <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-
node-match">
[a242]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a243]          /md:record
[a244]        </AttributeValue>
[a245]        <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
[a246]         DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a247]       </ResourceMatch>
[a248]      </Resource>
[a249]     </Resources>
[a250]     <Actions>
[a251]      <Action>
[a252]       <ActionMatch
[a253]        MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a254]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a255]           read
[a256]        </AttributeValue>
[a257]        <ActionAttributeDesignator
[a258]         AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a259]         DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a260]       </ActionMatch>
[a261]      </Action>
[a262]     </Actions>
[a263]    </Target>
[a264]    <Condition>
[a265]     <VariableReference VariableId="17590034"/>
[a266]    </Condition>
[a267]   </Rule>
[a268]  </Policy>
```

[a199] – [a201] Declaraciones del espacio de nombres XML.

[a205] Las expresiones XPath de la política se deben interpretar de acuerdo con W3C XPath:1999.

[a208] – [a221] Un elemento `<VariableDefinition>`. Define una función que evalúa la veracidad de la declaración: el atributo del sujeto número de paciente es igual al atributo número de paciente en el recurso.

[a209] El atributo `FunctionId` designa la función que se va a utilizar para la comparación. En el caso que nos ocupa, la comparación se realiza con la función "`urn:oasis:names:tc:xacml:1.0:function:string-equal`"; esta función se aplica a dos argumentos de tipo "http://www.w3.org/2001/XMLSchema#string".

[a210] El primer argumento de la definición de la variable es una función especificada por el atributo `FunctionId`. Dado que la función "`urn:oasis:names:tc:xacml:1.0:function:string-equal`" se aplica a argumentos de tipo "http://www.w3.org/2001/XMLSchema#string" y que `SubjectAttributeDesignator` selecciona un multiconjunto de tipo "http://www.w3.org/2001/XMLSchema#string", se utiliza la función "`urn:oasis:names:tc:xacml:1.0:function:string-one-and-only`". Esta función garantiza que el resultado de la evaluación de su argumento será un multiconjunto que contenga exactamente un valor.

[a211] El `SubjectAttributeDesignator` selecciona un multiconjunto de valores para el atributo del sujeto número de paciente en el contexto de petición.

[a215] El segundo argumento de la definición de la variable es una función especificada por el atributo FunctionId. Dado que la función "urn:oasis:names:tc:xacml:1.0:function:string-equal" se aplica a argumentos de tipo "http://www.w3.org/2001/XMLSchema#string" y que AttributeSelector selecciona un multiconjunto de tipo "http://www.w3.org/2001/XMLSchema#string", se utiliza la función "urn:oasis:names:tc:xacml:1.0:function:string-one-and-only". Esta función garantiza que el resultado de la evaluación de su argumento será un multiconjunto que contenga exactamente un valor.

[a216] El elemento <AttributeSelector> selecciona un multiconjunto de valores del contexto de petición utilizando una expresión XPath libre. En el caso que nos ocupa, selecciona el valor del número de paciente en el recurso. Obsérvese que los prefijos del espacio de nombres de la expresión XPath se determinan con las declaraciones estándar del espacio de nombres XML.

[a223] Identificador de la regla.

[a224] Declaración de efecto de la regla. Si el resultado de la evaluación de la regla es "Verdadero", emite el valor del atributo del Effect. Ese valor se combina entonces con los valores del Effect de otras reglas de acuerdo con el algoritmo de combinación de reglas.

[a225] – [a229] Descripción libre de la regla.

[a230] – [a263] Un objetivo de la regla define un conjunto de peticiones de decisión que se van a evaluar conforme a la regla. En este ejemplo se omiten los elementos <Subjects> y <Environments>.

[a231] – [a249] El elemento <Resources> contiene una secuencia disyuntiva de elementos <Resource>. En este ejemplo, sólo hay uno.

[a232] – [a248] El elemento <Resource> encierra la secuencia conjuntiva de elementos ResourceMatch. En este ejemplo, hay dos.

[a233] – [a240] El primer elemento <ResourceMatch> compara sus elementos derivados primero y segundo de acuerdo con la función de correspondencia. Habrá concordancia si el valor del primer argumento concuerda con alguno de los valores seleccionados por el segundo argumento. Esta correspondencia compara el espacio de nombres del objetivo del documento solicitado con el valor de "urn:example:med:schemas:record".

[a233] El atributo MatchId designa la función de correspondencia.

[a235] Valor literal del atributo con el que ha de haber concordancia.

[a237] – [a239] El elemento <ResourceAttributeDesignator> selecciona el espacio de nombres del objetivo a partir del recurso contenido en el contexto de petición. El nombre del atributo se especifica mediante el AttributeId.

[a241] – [a247] El segundo elemento <ResourceMatch>. Esta correspondencia compara el resultado de dos expresiones XPath. La segunda expresión XPath es el camino de ubicación hacia el elemento XML solicitado y la primera expresión XPath es el valor literal "/md:record". La función "xpath-node-match" devuelve "Verdadero" en la evaluación si el elemento XML está en un nivel inferior al elemento "/md:record".

[a250] – [a262] El elemento <Actions> contiene una secuencia disyuntiva de elementos <Action>. En el caso que nos ocupa, sólo hay un elemento <Action>.

[a251] – [a261] El elemento <Action> contiene una secuencia conjuntiva de elementos <ActionMatch>. En el caso que nos ocupa, sólo hay un elemento <ActionMatch>.

[a252] – [a260] El elemento <ActionMatch> compara sus elementos derivados primero y segundo de acuerdo con la función de correspondencia. Habrá concordancia si el valor del primer argumento concuerda con alguno de los valores seleccionados por el segundo argumento. En el caso que nos ocupa, el valor del atributo action-id del contexto de petición se compara con el valor literal "read" (leer).

[a264] – [a266] El elemento <Condition>. La regla sólo es aplicable si el resultado de la evaluación de una condición es "Verdadero". Esta condición contiene una referencia a una definición de variable codificada en otro elemento de la política.

II.2.4.2 Regla 2

La regla 2 ilustra el uso de una función matemática, es decir, el elemento <Apply> con FunctionId "urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration" para calcular la fecha en que el paciente cumple 16 años. También muestra el uso de expresiones de predicado, con FunctionId "urn:oasis:names:tc:xacml:1.0:function:and". En este ejemplo hay una función incorporada en el elemento <Condition> y otra referenciada en un elemento <VariableDefinition>.

```
[a269]  <?xml version="1.0" encoding="UTF-8"?>
[a270]  <Policy
[a271]  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
[a272]  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
[a273]  xmlns:xf="urn:oasis:names:tc:xacml:2.0:data-types"
[a274]  xmlns:md="http:www.med.example.com/schemas/record.xsd"
[a275]  PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:2"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-
overrides">
[a276]   <PolicyDefaults>
[a277]    <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-
19991116</XPathVersion>
[a278]   </PolicyDefaults>
[a279]   <Target/>
[a280]   <VariableDefinition VariableId="17590035">
[a281]    <Apply FunctionId="urn:oasis:names:tc:xacml:2.0:function:date-less-or-
equal">
[a282]     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-
only">
[a283]      <EnvironmentAttributeDesignator
[a284]      AttributeId= "urn:oasis:names:tc:xacml:1.0:environment:current-date"
[a285]      DataType="http://www.w3.org/2001/XMLSchema#date"/>
[a286]     </Apply>
[a287]     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-add-
yearMonthDuration">
[a288]      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-
and-only">
[a289]       <AttributeSelector RequestContextPath=
[a290]       "//md:record/md:patient/md:patientDoB/text()"
[a291]       DataType="http://www.w3.org/2001/XMLSchema#date"/>
[a292]      </Apply>
[a293]      <AttributeValue
[a294]      DataType="urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration">
[a295]       <xf:dt-yearMonthDuration>
[a296]        P16Y
[a297]       </xf:dt-yearMonthDuration>
[a298]      </AttributeValue>
[a299]     </Apply>
[a300]    </Apply>
[a301]   </VariableDefinition>
[a302]   <Rule
[a303]   RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:2"
[a304]   Effect="Permit">
[a305]    <Description>
[a306]     A person may read any medical record in the
[a307]     http://www.med.example.com/records.xsd namespace
[a308]     for which he or she is the designated parent or guardian,
[a309]     and for which the patient is under 16 years of age
[a310]    </Description>
[a311]    <Target>
[a312]     <Resources>
[a313]      <Resource>
[a314]       <ResourceMatch
[a315]       MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a316]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a317]           http://www.med.example.com/schemas/record.xsd
[a318]         </AttributeValue>
[a319]        <ResourceAttributeDesignator AttributeId=
"urn:oasis:names:tc:xacml:2.0:resource:target-namespace"
[a320]         DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a321]       </ResourceMatch>
[a322]       <ResourceMatch
[a323]       MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
[a324]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a325]          /md:record
[a326]        </AttributeValue>
[a327]        <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
```

```
[a328]         DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a329]        </ResourceMatch>
[a330]       </Resource>
[a331]      </Resources>
[a332]      <Actions>
[a333]       <Action>
[a334]        <ActionMatch
[a335]        MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a336]         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a337]           read
[a338]         </AttributeValue>
[a339]         <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a340]         DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a341]       </ActionMatch>
[a342]       </Action>
[a343]      </Actions>
[a344]    </Target>
[a345]    <Condition>
[a346]    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
[a347]     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
[a348]      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-
and-only">
[a349]       <SubjectAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:
[a350]  parent-guardian-id"
[a351]        DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a352]       </Apply>
[a353]       <Apply
[a354]       FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-
only">
[a355]        <AttributeSelector
[a356]        RequestContextPath="//xacml-context:Resource/xacml-
context:ResourceContent/md:record/md:parentGuardian/md:parentGuardianId/text()"
[a357]        DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a358]        </Apply>
[a359]       </Apply>
[a360]       <VariableReference VariableId="17590035"/>
[a361]     </Apply>
[a362]    </Condition>
[a363]   </Rule>
[a364]  </Policy>
```

[a280] – [a301] El elemento `<VariableDefinition>` contiene parte de la condición (¿Es el paciente menor de 16 años?). El paciente es menor de 16 años si el valor para la fecha actual es menor que el que resulta de añadir 16 a la fecha de nacimiento del paciente.

[a281] – [a300] La función "`urn:oasis:names:tc:xacml:1.0:function:date-less-or-equal`" se utiliza para calcular la diferencia de dos argumentos de fecha.

[a282] – [a286] El primer argumento de fecha utiliza "`urn:oasis:names:tc:xacml:1.0:function:date-one-and-only`" para garantizar que el multiconjunto de valores seleccionados mediante su argumento contiene exactamente un valor de tipo "http://www.w3.org/2001/XMLSchema#date".

[a284] La fecha actual se evalúa seleccionando el atributo del entorno "`urn:oasis:names:tc:xacml:1.0:environment:current-date`".

[a287] – [a299] El segundo argumento de fecha utiliza "`urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration`" para calcular la fecha en que el paciente cumple 16 años, sumando 16 años a la fecha de nacimiento del paciente. El primero de sus argumentos es de tipo "http://www.w3.org/2001/XMLSchema#date", y el segundo de tipo "`urn:oasis:names:tc:xacml:2.0:data-types:yearMonthDuration`".

[a289] El elemento `<AttributeSelector>` selecciona la fecha de nacimiento del paciente aplicando la expresión XPath al contenido del recurso.

[a293] – [a298] Periodo de 16 años indicado mediante año/mes.

[a311] – [a344] Declaración y objetivo de la regla. En la regla 1 de II.4.2.4.1 se puede encontrar una explicación detallada de estos elementos.

[a345] – [a362] El elemento <Condition>. La regla sólo es aplicable si el resultado de la evaluación de la condición es "Verdadero". Esta condición evalúa la veracidad de la declaración: el solicitante es el progenitor o tutor designado y el paciente es menor de 16 años. Contiene un elemento <Apply> incorporado y un elemento <VariableDefinition> referenciado.

[a346] La condición utiliza la función "urn:oasis:names:tc:xacml:1.0:function:and". Se trata de una función booleana que se aplica a uno o más argumentos booleanos (dos en este caso) y ejecuta la operación lógica "AND" para determinar la veracidad de la expresión.

[a347] – [a359] Se evalúa la primera parte de la condición (¿Es el solicitante el progenitor o el tutor designado?). La función es "urn:oasis:names:tc:xacml:1.0:function:string-equal" y tiene dos argumentos de tipo "http://www.w3.org/2001/XMLSchema#string".

[a348] Designa el primer argumento. Dado que "urn:oasis:names:tc:xacml:1.0:function:string-equal" tiene argumentos de tipo "http://www.w3.org/2001/XMLSchema#string", se utiliza la función "urn:oasis:names:tc:xacml:1.0:function:string-one-and-only" para comprobar si el atributo del sujeto "urn:oasis:names:tc:xacml:2.0:example:attribute:parent-guardian-id" del contexto de petición contiene exactamente un valor.

[a353] Designa el segundo argumento. El valor del atributo del sujeto "urn:oasis:names:tc:xacml:2.0:example:attribute:parent-guardian-id" se selecciona a partir del contexto de petición utilizando el elemento <SubjectAttributeDesignator>.

[a354] Al igual que antes, la función "urn:oasis:names:tc:xacml:1.0:function:string-one-and-only" se utiliza para garantizar que el multiconjunto de valores seleccionado mediante su argumento contiene exactamente un valor de tipo "http://www.w3.org/2001/XMLSchema#string".

[a355] El segundo argumento selecciona el valor del elemento <md:parentGuardianId> a partir del contenido del recurso utilizando el elemento <AttributeSelector>. Este elemento contiene una expresión XPath libre que señala el contexto de petición. Obsérvese que todos los prefijos del espacio de nombres de la expresión XPath se determinan con declaraciones normalizadas del espacio de nombres. El resultado de la evaluación de AttributeSelector es un multiconjunto de valores de tipo "http://www.w3.org/2001/XMLSchema#string".

[a360] Hace referencia al elemento <VariableDefinition> que define la segunda parte de la condición.

II.2.4.3 Regla 3

La regla 3 ilustra el uso de una obligación. La sintaxis del elemento <Rule> de XACML no incluye un elemento adecuado para transportar una obligación, por lo que es necesario formatear la regla 3 como elemento <Policy>.

```
[a365] <?xml version="1.0" encoding="UTF-8"?>
[a366] <Policy
[a367]  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
[a368]  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
[a369]  xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
[a370] xmlns:md="http:www.med.example.com/schemas/record.xsd"
[a371]  PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:3"
[a372]  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
[a373]  <Description>
[a374]   Policy for any medical record in the
[a375]   http://www.med.example.com/schemas/record.xsd namespace
[a376]  </Description>
[a377]  <PolicyDefaults>
[a378]   <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-19991116</XPathVersion>
[a379]  </PolicyDefaults>
[a380]  <Target>
[a381]   <Resources>
[a382]    <Resource>
[a383]     <ResourceMatch
[a384]     MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a385]      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a386]       urn:example:med:schemas:record
[a387]      </AttributeValue>
[a388]      <ResourceAttributeDesignator AttributeId=
[a389]      "urn:oasis:names:tc:xacml:1.0:resource:target-namespace"
[a390]      DataType="http://www.w3.org/2001/XMLSchema#string"/>
```

```
[a391]     </ResourceMatch>
[a392]    </Resource>
[a393]   </Resources>
[a394]  </Target>
[a395]  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:3"
[a396]  Effect="Permit">
[a397]   <Description>
[a398]    A physician may write any medical element in a record
[a399]    for which he or she is the designated primary care
[a400]    physician, provided an email is sent to the patient
[a401]   </Description>
[a402]   <Target>
[a403]    <Subjects>
[a404]     <Subject>
[a405]      <SubjectMatch
[a406]      MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a407]       <AttributeValue  DataType="http://www.w3.org/2001/XMLSchema#string">
[a408]         physician
[a409]        </AttributeValue>
[a410]        <SubjectAttributeDesignator AttributeId=
"urn:oasis:names:tc:xacml:2.0:example:attribute:role"
[a411]         DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a412]      </SubjectMatch>
[a413]     </Subject>
[a414]    </Subjects>
[a415]    <Resources>
[a416]     <Resource>
[a417]      <ResourceMatch
[a418]      MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
[a419]       <AttributeValue
[a420]        DataType="http://www.w3.org/2001/XMLSchema#string">
[a421]          /md:record/md:medical
[a422]        </AttributeValue>
[a423]       <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
[a424]       DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a425]      </ResourceMatch>
[a426]     </Resource>
[a427]    </Resources>
[a428]    <Actions>
[a429]     <Action>
[a430]      <ActionMatch
[a431]      MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a432]       <AttributeValue
[a433]       DataType="http://www.w3.org/2001/XMLSchema#string">
[a434]         write
[a435]       </AttributeValue>
[a436]       <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a437]       DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a438]      </ActionMatch>
[a439]     </Action>
[a440]    </Actions>
[a441]   </Target>
[a442]   <Condition>
[a443]   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a444]    <Apply
[a445]    FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
[a446]     <SubjectAttributeDesignator
[a447]     AttributeId="urn:oasis:names:tc:xacml:2.0:example: attribute:physician-
id"
[a448]     DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a449]    </Apply>
[a450]    <Apply
[a451]    FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
[a452]     <AttributeSelector RequestContextPath=
[a453]     "//xacml-context:Resource/xacml-
context:ResourceContent/md:record/md:primaryCarePhysician/md:registrationID/text()"
[a454]     DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a455]    </Apply>
[a456]     </Apply>
```

```
[a457]   </Condition>
[a458]  </Rule>
[a459]  <Obligations>
[a460]   <Obligation
ObligationId="urn:oasis:names:tc:xacml:example:obligation:email"
[a461]   FulfillOn="Permit">
[a462]    <AttributeAssignment
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:mailto"
[a463]    DataType="http://www.w3.org/2001/XMLSchema#string">
[a464]     &lt;AttributeSelector RequestContextPath=
[a465]     "//md:/record/md:patient/md:patientContact/md:email"
[a466]     DataType="http://www.w3.org/2001/XMLSchema#string"/&gt ;
[a467]    </AttributeAssignment>
[a468]    <AttributeAssignment
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
[a469]    DataType="http://www.w3.org/2001/XMLSchema#string">
[a470]      Your medical record has been accessed by:
[a471]    </AttributeAssignment>
[a472]    <AttributeAssignment
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:text"
[a473]    DataType="http://www.w3.org/2001/XMLSchema#string">
[a474]     &lt;SubjectAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
[a475]     DataType="http://www.w3.org/2001/XMLSchema#string"/&gt;
[a476]    </AttributeAssignment>
[a477]   </Obligation>
[a478]  </Obligations>
[a479] </Policy>
```

[a366] – [a372] El elemento `<Policy>` incluye declaraciones normalizadas del espacio de nombres así como parámetros específicos de la política, como `PolicyId` o `RuleCombiningAlgId`.

[a371] Identificador de la política. Este parámetro permite que se haga referencia a la política en un conjunto de políticas.

[a372] El algoritmo de combinación de reglas determina el algoritmo para combinar los resultados de la evaluación de las reglas.

[a373] – [a376] Descripción de la política en formato libre.

[a379] – [a394] Objetivo de la política. El objetivo de la política define un conjunto de peticiones de decisión aplicables. El elemento `<Target>` en `<Policy>` tiene la misma estructura que el elemento `<Target>` en `<Rule>`. En el caso que nos ocupa, el objetivo de la política es el conjunto de todos los recursos XML que son conformes al espacio de nombres "`urn:example:med:schemas:record`".

[a395] El único elemento `<Rule>` incluido en esta `<Policy>`. En el encabezamiento de la regla se especifican dos parámetros: `RuleId` y `Effect`.

[a402] – [a441] El objetivo de la regla restringe el objetivo de la política.

[a405] – [a412] El elemento `<SubjectMatch>` limita a la regla a los sujetos cuyo atributo "`urn:oasis:names:tc:xacml:2.0:example:attribute:role`" sea "`physician`" (médico).

[a417] – [a425] El elemento `<ResourceMatch>` limita la regla a los recursos que se ajusten a la expresión XPath "`/md:record/md:medical`".

[a430] – [a438] El elemento `<ActionMatch>` limita la regla a las acciones cuyo atributo "`urn:oasis:names:tc:xacml:1.0:action:action-id`" sea "write" (escribir).

[a442] – [a457] El elemento `<Condition>`. La regla sólo es aplicable a la petición de decisión si el resultado de la evaluación de la condición es "Verdadero". Esta condición compara el valor del atributo del sujeto "`urn:oasis:names:tc:xacml:2.0:example:attribute:physician-id`" con el valor del elemento `<registrationId>` en el registro médico consultado.

[a459] – [a478] El elemento `<Obligations>`. Las obligaciones son un conjunto de operaciones que debe realizar el PEP cuando se toma una decisión de autorización. Una obligación puede estar asociada con una decisión de autorización "Permitir" o "Denegar". El elemento contiene una única obligación.

[a460] – [a477] El elemento `<Obligation>` consta del atributo `ObligationId`, del valor de la decisión de autorización que exige el cumplimiento de esa obligación y de un conjunto de asignaciones de atributos. El PDP no decide las asignaciones de atributos, lo hace el PEP.

[a460] El atributo `ObligationId` determina la obligación. En el caso que nos ocupa, consiste en que el PEP envíe un correo electrónico.

[a461] El atributo `FulfillOn` define el valor de la decisión de autorización que exige el cumplimiento de esa obligación. En el caso que nos ocupa, se trata del momento en que se permite el acceso.

[a462] – [a467] El primer parámetro indica al PEP dónde está la dirección de correo electrónico en el recurso.

[a468] – [a471] El segundo parámetro contiene texto literal para el cuerpo del correo electrónico.

[a472] – [a476] El tercer parámetro indica al PEP dónde está el texto adicional para el cuerpo del correo electrónico en el recurso.

II.2.4.4 Regla 4

La regla 4 ilustra el uso del valor de efecto "Denegar" y una `<Rule>` sin ningún elemento `<Condition>`.

```
[a480]  <?xml version="1.0" encoding="UTF-8"?>
[a481]  <Policy
[a482]  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
[a483]  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
[a484]  xmlns:md="http:www.med.example.com/schemas/record.xsd"
[a485]  PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:4"
[a486]  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:deny-overrides">
[a487]   <PolicyDefaults>
[a488]    <XPathVersion>http://www.w3.org/TR/1999/Rec-xpath-
19991116</XPathVersion>
[a489]   </PolicyDefaults>
[a490]   <Target/>
[a491]   <Rule
[a492]   RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:4"
[a493]   Effect="Deny">
[a494]    <Description>
[a495]     An Administrator shall not be permitted to read or write
[a496]     medical elements of a patient record in the
[a497]     http://www.med.example.com/records.xsd namespace.
[a498]    </Description>
[a499]    <Target>
[a500]     <Subjects>
[a501]      <Subject>
[a502]       <SubjectMatch
[a503]       MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a504]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a505]         administrator
[a506]        </AttributeValue>
[a507]        <SubjectAttributeDesignator AttributeId=
[a508]        "urn:oasis:names:tc:xacml:2.0:example:attribute:role"
[a509]        DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a510]       </SubjectMatch>
[a511]      </Subject>
[a512]     </Subjects>
[a513]     <Resources>
[a514]      <Resource>
[a515]       <ResourceMatch
[a516]       MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a517]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a518]         urn:example:med:schemas:record
[a519]        </AttributeValue>
[a520]        <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:target-namespace"
[a521]         DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a522]       </ResourceMatch>
[a523]       <ResourceMatch
[a524]       MatchId="urn:oasis:names:tc:xacml:1.0:function:xpath-node-match">
[a525]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a526]         /md:record/md:medical
```

```
[a527]        </AttributeValue>
[a528]        <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:xpath"
[a529]        DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a530]       </ResourceMatch>
[a531]      </Resource>
[a532]     </Resources>
[a533]     <Actions>
[a534]      <Action>
[a535]       <ActionMatch
[a536]       MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a537]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a538]         read
[a539]        </AttributeValue>
[a540]        <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a541]        DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a542]       </ActionMatch>
[a543]      </Action>
[a544]      <Action>
[a545]       <ActionMatch
[a546]       MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a547]        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
[a548]         write
[a549]        </AttributeValue>
[a550]        <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
[a551]        DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a552]       </ActionMatch>
[a553]      </Action>
[a554]     </Actions>
[a555]    </Target>
[a556]   </Rule>
[a557]  </Policy>
```

[a492] – [a493] La declaración del elemento `<Rule>`.

[a493] Regla `Effect`. Cuando el resultado es "Verdadero", el valor de la regla será siempre su efecto. Esta regla `Effect` es "Denegar", lo que significa que, según esta regla, se debe denegar el acceso cuando el resultado de su evaluación sea "Verdadero".

[a494] – [a498] Descripción de la regla en formato libre.

[a499] – [a555] Objetivo de la regla. El objetivo de la regla define el conjunto de peticiones de decisión que son aplicables a la regla.

[a502] – [a510] El elemento `<SubjectMatch>` limita la regla a los sujetos cuyo atributo "`urn:oasis:names:tc:xacml:2.0:example:attribute:role`" sea "administrator" (administrador).

[a513] – [a532] El elemento `<Resources>` contiene un elemento `<Resource>` que, a su vez, contiene dos elementos `<ResourceMatch>`. Habrá concordancia del objetivo si el recurso identificado por el contexto de petición se ajusta a ambos criterios de correspondencia.

[a515] – [a522] El primer elemento `<ResourceMatch>` limita la regla a los recursos cuyo atributo "`urn:oasis:names:tc:xacml:2.0:resource:target-namespace`" sea "`urn:example:med:schemas:record`".

[a523] – [a530] El segundo elemento `<ResourceMatch>` limita la regla a los elementos XML que concuerdan con la expresión XPath "`/md:record/md:medical`".

[a533] – [a554] El elemento `<Actions>` contiene dos elementos `<Action>`, cada uno de los cuales contiene un elemento `<ActionMatch>`. Habrá concordancia del objetivo si la acción identificada en el contexto de petición se ajusta a alguno de los criterios de correspondencia.

[a535] – [a552] El elemento `<ActionMatch>` limita la regla a las acciones cuyo atributo "`urn:oasis:names:tc:xacml:1.0:action:action-id`" sea "read" (leer) o "write" (escribir).

Esta regla no tiene ningún elemento `<Condition>`.

II.2.4.5 Ejemplo de conjunto de políticas

En esta cláusula se utilizan los ejemplos de las cláusulas anteriores para ilustrar el proceso de combinación de políticas. La política que rige el acceso a la lectura de elementos médicos de un registro se forma a partir de cada una de las cuatro reglas descritas en II.4.2.3. En lenguaje explícito, la regla combinada es:

- o bien el solicitante es el paciente; o bien
- el solicitante es el progenitor o el tutor y el paciente es menor de 16 años; o bien
- el solicitante es el médico de atención primaria y se envía una notificación al paciente; y
- el solicitante no es un administrador.

El siguiente conjunto de políticas ilustra la combinación de políticas. La política 3 se incluye por referencia y la política 2 se incluye explícitamente.

```
[a558]    <?xml version="1.0" encoding="UTF-8"?>
[a559]     <PolicySet
[a560]     xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
[a561]     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-schema-os.xsd"
[a562]     PolicySetId=
[a563]     "urn:oasis:names:tc:xacml:2.0:example:policysetid:1"
[a564]     PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:
[a565]     policy-combining-algorithm:deny-overrides">
[a566]      <Description>
[a567]       Example policy set.
[a568]      </Description>
[a569]      <Target>
[a570]       <Resources>
[a571]        <Resource>
[a572]         <ResourceMatch
[a573]         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
[a574]          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
[a575]           urn:example:med:schema:records
[a576]          </AttributeValue>
[a577]          <ResourceAttributeDesignator AttributeId=
[a578]          "urn:oasis:names:tc:xacml:1.0:resource:target-namespace"
[a579]          DataType="http://www.w3.org/2001/XMLSchema#string"/>
[a580]         </ResourceMatch>
[a581]        </Resource>
[a582]       </Resources>
[a583]      </Target>
[a584]     <PolicyIdReference>
[a585]      urn:oasis:names:tc:xacml:2.0:example:policyid:3
[a586]     </PolicyIdReference>
[a587]     <Policy
[a588]     PolicyId="urn:oasis:names:tc:xacml:2.0:example:policyid:2"
[a589]     RuleCombiningAlgId=
[a590]    "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
[a591]      <Target/>
[a592]      <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:1"
[a593]      Effect="Permit">
[a594]      </Rule>
[a595]      <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:2"
[a596]      Effect="Permit">
[a597]      </Rule>
[a598]      <Rule RuleId="urn:oasis:names:tc:xacml:2.0:example:ruleid:4"
[a599]      Effect="Deny">
[a600]      </Rule>
[a601]     </Policy>
[a602]    </PolicySet>
```

[a559] – [a565] La declaración del elemento `<PolicySet>`. Se incluyen declaraciones estándar del espacio de nombres XML.

[a562] El atributo `PolicySetId` se utiliza para identificar este conjunto de políticas para su posible inclusión en otro conjunto de políticas.

[a564] El identificador del algoritmo de combinación de políticas. Las políticas y los conjuntos de políticas de este conjunto se combinan de acuerdo con el algoritmo de combinación de políticas especificado cuando se calcula la decisión de autorización.

[a566] – [a568] Descripción del conjunto de políticas en formato libre.

[a569] – [a583] El elemento `<Target>` del conjunto de políticas define el conjunto de peticiones de decisión que son aplicables a este elemento `<PolicySet>`.

[a584] `PolicyIdReference` incluye una política mediante su identificador.

[a588] La `Policy 2` se incluye explícitamente en este conjunto de políticas. Para simplificar, no se han incluido las reglas de la `Policy 2`.

Apéndice III

Ejemplos de funciones de multiconjunto de orden superior

III.1 Ejemplos de funciones de multiconjunto de orden superior

En el presente apéndice se describen funciones en XACML que efectúan operaciones sobre multiconjuntos y que se puedan aplicar a los multiconjuntos en general.

A modo de ejemplo, un lenguaje funcional de aplicación general llamado Haskell (véase [Haskell]) se utiliza para especificar la semántica de esas funciones. Aunque la descripción en inglés es adecuada, resulta útil una correcta especificación de la semántica.

De forma resumida, en la siguiente notación Haskell se define una función aplicando cláusulas a modelos de estructuras, principalmente listas. El símbolo "[]" señala la lista vacía, mientras que la expresión "(x:xs)" se compara con un argumento de una lista no vacía en la que "x" representa el primer elemento de la lista, y "xs", el resto de la lista, que podría estar vacía. Utilizamos el concepto Haskell de lista como conjunto ordenado de elementos para definir los multiconjuntos de valores XACML.

Ésta sería una definición Haskell simple de una función conocida "urn:oasis:names:tc:xacml:1.0:function:and" que se aplica a una lista de valores de tipo booleano:

```
and:: [Bool]    -> Bool
and []          = True
and (x:xs)      = x && (and xs)
```

La primera línea de la definición indicada mediante "::" describe formalmente el tipo de datos de la función; se aplica a una lista de booleanos, señalados mediante "[Bool]", y devuelve un booleano, señalado mediante "Bool". La segunda línea de la definición es una cláusula que establece que el resultado de la función "and" aplicada a la lista vacía es "Verdadero". La tercera línea de la definición es una cláusula que establece que para una lista no vacía, siendo su primer elemento "x" un valor de tipo de datos booleano, la función "and" aplicada a x se combinará con el resultado de aplicar de forma recursiva la función "and" al resto de la lista utilizando la función conjuntiva lógica, indicada mediante el símbolo infijo "&&". Por supuesto, el resultado de aplicación de la función "and" será "Verdadero" si y sólo si la lista a la que se aplica la función está vacía o todos los elementos de la lista son "Verdadero". Por ejemplo, el resultado de la evaluación de las siguientes expresiones Haskell,

```
(and []), (and [True]), (and [True,True]), (and [True,True,False])
```

será respectivamente "Verdadero", "Verdadero", "Verdadero" y "Falso".

1) `urn:oasis:names:tc:xacml:1.0:function:any-of`

   La semántica de esta operación en notación Haskell es:

```
any_of :: ( a -> b -> Bool )    -> a -> [b] -> Bool
any_of   f   a   []             = False
any_of   f   a   (x:xs)         = (f  a  x) || (any_of  f  a  xs)
```

   En la notación anterior, "f" es la función que se va a aplicar, "a" el valor primitivo y "(x:xs)" representa el primer elemento de la lista por "x" y el resto de la lista por "xs".

Por ejemplo, la siguiente expresión devolverá "Verdadero":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
      <Function
            FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            Paul
      </AttributeValue>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                  John
            </AttributeValue>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                  Paul
            </AttributeValue>
            <AttributeValue
                  DataType="http://www.w3.org/2001/XMLSchema#string">George
            </AttributeValue>
            <AttributeValue
                  DataType="http://www.w3.org/2001/XMLSchema#string">
                  Ringo
            </AttributeValue>
      </Apply>
</Apply>
```

El resutado de esta expresión es "Verdadero" porque el primer argumento es igual a uno o más de los elementos del multiconjunto, de acuerdo con la función.

2) `urn:oasis:names:tc:xacml:1.0:function:all-of`

La semántica de esta operación en notación Haskell es:

```
all_of :: ( a -> b -> Bool )  -> a -> [b] -> Bool
all_of   f   a   []                 = True
all_of   f   a   (x:xs)             = (f  a  x) && (all_of  f  a  xs)
```

En la notación anterior, "f" es la función que se va a aplicar, "a" es el valor primitivo y "(x:xs)" representa el primer elemento de la lista por "x" y el resto de la lista por "xs".

Por ejemplo, el resultado de la evaluación de la siguiente expresión será "Verdadero":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:all-of">
  <Function FunctionId="urn:oasis:names:tc:xacml:2.0:function:integer-greater"/>
  <AttributeValueDataType="http://www.w3.org/2001/XMLSchema#integer">10</
AttributeValue>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
        <AttributeValue
  DataType="http://www.w3.org/2001/XMLSchema#integer">9</AttributeValue>
        <AttributeValue
  DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
        <AttributeValue
  DataType="http://www.w3.org/2001/XMLSchema#integer">4</AttributeValue>
        <AttributeValue
  DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
  </Apply>
</Apply>
```

El resultado de esta expresión es "Verdadero" porque el primer argumento (10) es mayor que todos los elementos del multiconjunto (9, 3, 4 y 2).

3) `urn:oasis:names:tc:xacml:1.0:function:any-of-any`

Partiendo de la función "any_of" definida anteriormente, la semántica de la función "any_of_any" en notación Haskell es:

```
any_of_any :: ( a -> b -> Bool )    -> [a]-> [b] -> Bool
any_of_any  f  []          ys       = False
any_of_any  f  (x:xs)      ys       = (any_of  f  x  ys) || (any_of_any  f  xs  ys)
```

En la notación anterior, "f" es la función que se va a aplicar y "(x:xs)" representa el primer elemento de la lista por "x" y el resto de la lista por "xs".

Por ejemplo, el resultado de la evaluación de la siguiente expresión será "Verdadero":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
  <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Ringo</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Mary</AttributeValue>
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">John</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Paul</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">George</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Ringo</AttributeValue>
  </Apply>
</Apply>
```

El resultado de esta expresión es "Verdadero" porque al menos uno de los elementos del primer multiconjunto, en concreto "Ringo", es igual a uno o más de los elementos del segundo.

4) `urn:oasis:names:tc:xacml:1.0:function:all-of-any`

Partiendo de la función "any_of" que ya se ha definido en Haskell, la semántica de la función "all_of_any" en notación Haskell es:

```
all_of_any :: ( a -> b -> Bool )          -> [a]-> [b] -> Bool
all_of_any   f  []          ys                    = True
all_of_any   f  (x:xs)      ys                    = (any_of  f  x  ys) && (all_of_any f  xs  ys)
```

En la notación anterior, "f" es la función que se va a aplicar y "(x:xs)" representa el primer elemento de la lista por "x" y el resto de la lista por "xs".

Por ejemplo, el resultado de la evaluación de la siguiente expresión será "Verdadero":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:all-of-any">
  <Function FunctionId="urn:oasis:names:tc:xacml:2.0:function:integer-greater"/>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">10</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">20</AttributeValue>
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">5</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">19</AttributeValue>
  </Apply>
</Apply>
```

El resultado de esta expresión es "Verdadero" porque cada uno de los elementos del primer multiconjunto es mayor que uno o más de los elementos del segundo.

5) `urn:oasis:names:tc:xacml:1.0:function:any-of-all`

Partiendo de la función "all_of" que ya se ha definido en Haskell, la semántica de la función "any_of_all" en notación Haskell es:

```
any_of_all :: ( a -> b -> Bool )          -> [a]-> [b] -> Bool
any_of_all  f  []           ys                    = False
any_of_all  f  (x:xs)       ys                    = (all_of  f  x  ys) || ( any_of_all  f  xs  ys)
```

En la notación anterior, "f" es el nombre de la función que se va a aplicar y "(x:xs)" representa el primer elemento de la lista por "x" y el resto de la lista por "xs".

Por ejemplo, el resultado de la evaluación de la siguiente expresión será "Verdadero":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-all">
  <Function FunctionId="urn:oasis:names:tc:xacml:2.0:function:integer-greater"/>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">5</AttributeValue>
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">4</AttributeValue>
  </Apply>
</Apply>
```

El resultado de esta expresión es "Verdadero" porque, para todos los valores del segundo multiconjunto, hay un valor del primero que es mayor.

6) `urn:oasis:names:tc:xacml:1.0:function:all-of-all`

Partiendo de la función "all_of" que ya se ha definido en Haskell, la semántica de la función "all_of_all" en notación Haskell es:

```
all_of_all :: ( a -> b -> Bool )        -> [a] -> [b] -> Bool

all_of_all   f  []          ys              = True

all_of_all   f  (x:xs)      ys              = (all_of  f  x  ys) && (all_of_all  f  xs  ys)
```

En la notación anterior, "f" es la función que se va a aplicar y "(x:xs)" representa el primer elemento de la lista por "x" y el resto de la lista por "xs".

Por ejemplo, el resultado de la evaluación de la siguiente expresión será "Verdadero":

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:all-of-all">
  <Function FunctionId="urn:oasis:names:tc:xacml:2.0:function:integer-greater"/>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">6</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">5</AttributeValue>
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-bag">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">1</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">3</AttributeValue>
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">4</AttributeValue>
  </Apply>
</Apply>
```

El resultado de esta expresión es "Verdadero" porque todos los elementos del primer multiconjunto, "5" y "6", son mayores que todos los valores enteros "1", "2", "3" y "4" del segundo.

7) `urn:oasis:names:tc:xacml:1.0:function:map`

En notación Haskell, esta función se define así:

```
map:: (a -> b)  -> [a] -> [b]

map  f  []          = []

map  f  (x:xs)      =  (f x) : (map  f   xs)
```

En la notación anterior, "f" es la función que se va a aplicar y "(x:xs)" representa el primer elemento de la lista por "x" y el resto de la lista por "xs".

Por ejemplo, el resultado de la evaluación de la siguiente expresión,

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:map">
  <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-normalize-
to-lower-case">
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
     <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Hello</AttributeValue>
     <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">World!</AttributeValue>
  </Apply>
</Apply>
```

será un multiconjunto que contenga "hello" y "world!".

BIBLIOGRAFÍA

[Haskell] THOMPSON (S.): Haskell: The Craft of Functional Programming (2nd Edition), *Addison Wesley*, ISBN 0-201-34275-8, 1996.

[IEEE 754] IEEE 754-1985, *Binary Floating-Point Arithmetic*, ISBN 1-5593-7653-8, IEEE Product No. SH10116-TBR.

[RBAC] ANSI INCITS 359-2004, *Information technology – Role Based Access Control*, http://csrc.nist.gov/rbac/.

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
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 multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
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, comunicaciones de sistemas abiertos y seguridad**
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Impreso en Suiza
Ginebra, 2007