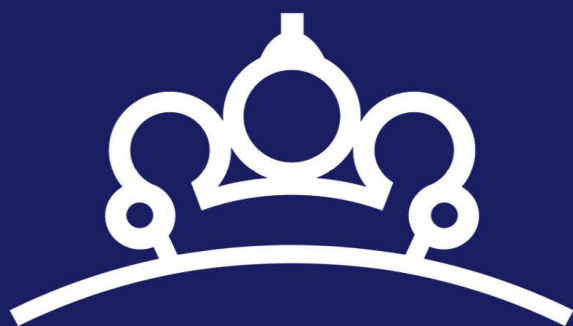


PUBLICACIONES TÉCNICAS

**BIBLIOTECA
NACIONAL
DE ESPAÑA**



BNE

Diccionario de datos PREMIS



DICCIONARIO DE DATOS PREMIS
De Metadatos de Preservación
Versión 2.0

DICCIONARIO DE DATOS PREMIS

De Metadatos de Preservación

Versión 2.0

Traducción: Lorea Elduayen Pereda y Bárbara Guiomar Muñoz de Solano y Palacios.

© Biblioteca Nacional de España. NIPO 552-10-015-3

Esta traducción ha sido publicada con la autorización de la Library of Congress, organismo competente para el mantenimiento de PREMIS.

Página oficial: <http://www.loc.gov/standards/premis>

Catálogo general de publicaciones oficiales de la Administración General del Estado:
<http://publicacionesoficiales.boe.es>

ÍNDICE

1. INTRODUCCIÓN.....	6
1.1 Antecedentes.....	6
1.1.1 Desarrollo del Diccionario de Datos PREMIS original	8
1.1.2 Metadatos fundamentales de preservación aplicables.....	9
1.2 El modelo de datos PREMIS.....	11
1.2.1 Más información sobre Objetos.....	14
1.2.2 Entidades Intelectuales y Objetos	17
1.2.3 Más información sobre Acontecimientos.....	18
1.2.4 Más información sobre Agentes	21
1.2.5 Más información sobre Derechos	21
1.3 Cuestiones generales sobre la estructura y el uso del Diccionario de Datos.....	22
1.3.1 Identificadores.....	22
1.3.2 Relaciones entre Objetos.....	24
1.3.3 Relaciones entre entidades de diferentes tipos.....	26
1.3.4 El principio 1:1	26
1.4 Consideraciones relativas a la implementación	27
1.4.1 Conformidad con PREMIS	27
1.4.2 Implementación del modelo de datos	29
1.4.3 Almacenamiento de metadatos.....	30
1.4.4 Provisión de los valores de los metadatos	31
1.4.5 Capacidad de extensión.....	33
1.4.6 Los formatos de fecha y hora en PREMIS	35
2. EL DICCIONARIO DE DATOS PREMIS VERSIÓN 2.0.....	37
2.1. Alcance del Diccionario de Datos.....	39
2.2. Entidad Objeto	42
2.2.1. Tipos de Entidad	42
2.2.2. Propiedades de la Entidad	42
2.2.3. Unidades semánticas de la Entidad	42
2.3. Entidad Acontecimiento	145
2.3.1. Propiedades de la Entidad	145
2.3.2. Unidades semánticas de la Entidad	145
2.4. Entidad Agentes	167
2.4.1. Propiedades de la Entidad	167
2.4.2. Unidades semánticas de la Entidad	167
2.5. Entidad Derechos	173
2.5.1. Propiedades de la Entidad	173
2.5.2. Unidades Semánticas de la Entidad.....	173
3. TEMAS ESPECIALES	211
3.1. Información de formato.....	211
3.1.1. Entorno	214
3.1.2. Características y nivel de composición de los objetos: el modelo «cebolla»	217
3.1.3. Fijeza, integridad y autenticidad.....	219
3.1.4. Firmas digitales	220
3.1.5. Metadatos no fundamentales	225
4. METODOLOGÍA	231

5. GLOSARIO 232

AGRADECIMIENTOS

Miembros del Comité Editorial PREMIS

Rebecca Guenther, Library of Congress, Chair

Steve Bordwell, General Register Office for Scotland

Olaf Brandt, Koninklijke Bibliotheek, Netherlands

Priscilla Caplan, Florida Center for Library Automation

Gerard Clifton, National Library of Australia

Angela Dappert, British Library

Markus Enders, Staats- und Universitätsbibliothek Göttingen/British Library

Brian Lavoie, OCLC

Bill Leonard, Library and Archives Canada

Zhiwu Xie, Los Alamos National Laboratory

Agradecimientos especiales

Estas personas han contribuido con la experiencia que les aportó su pertenencia al Comité Editorial PREMIS.

Rory McLeod, British Library

Yaniv Levi, ExLibris

Estas personas constituyeron el Grupo de Trabajo PREMIS original (Preservation Metadata: Implementation Strategies), es decir, el grupo de trabajo encargado de desarrollar la primera versión del Diccionario de Datos

Priscilla Caplan, Florida Center for Library Automation, co-chair

Rebecca Guenther, Library of Congress, co-chair

Robin Dale, RLG liaison

Brian Lavoie, OCLC liaison

George Barnum, U.S. Government Printing Office

Charles Blair, University of Chicago

Olaf Brandt, Göttingen State and University Library

Mikki Carpenter, Museum of Modern Art

Adam Farquhar, British Library

David Gewirtz, Yale University

Keith Glavash, MIT/DSpace

Andrea Goethals, Florida Center for Library Automation

Cathy Hartman, University of North Texas

Helen Hodgart, British Library

Nancy Hoebelheinrich, Stanford University

Roger Howard, J. Paul Getty Museum

Sally Hubbard, Getty Research Institute

Mela Kircher, OCLC

John Kunze, California Digital Library

Vicky McCargar, Los Angeles Times

Jerome McDonough, New York University/METS

Evan Owens, Ithaka-Electronic Archiving Initiative

Erin Rhodes, U.S. National Archives and Records Administration

Madi Solomon, Walt Disney Corporation

Angela Spinazze, ATSPIN Consulting

Stefan Strathmann, Göttingen State and University Library

Günter Waibel, RLG

Lisa Weber, U.S. National Archives and Records Administration

Robin Wendler, Harvard University

Hilde van Wijngaarden, National Library of the Netherlands

Andrew Wilson, National Archives of Australia and British Library

Deborah Woodyard-Robinson, British Library and Woodyard-Robinson Holdings Ltd.

SITIO WEB PREMIS Y CORREO ELECTRÓNICO

Sitio web relacionado con el mantenimiento de los metadatos PREMIS:
<http://www.loc.gov/standards/premis/>.

Lista de discusión del Grupo de implementadores PREMIS: pig@loc.gov. Para suscribirse remitir correo a: listserv@loc.gov con el mensaje "subscribe pig [su nombre]"

Para la remisión de preguntas y comentarios remítase a: premis@loc.gov.

1. INTRODUCCIÓN

1.1 Antecedentes

En junio del 2003, OCLC (*Online Computer Library Center*) y RLG (*Research Libraries Group*) patrocinaron la formación del grupo de trabajo PREMIS (por sus siglas en inglés *Preservation Metadata: Implementation Strategies*, Metadatos de preservación: estrategias de ejecución), compuesto por expertos internacionales en la utilización de metadatos aplicados a actividades de preservación digital. El grupo de trabajo estaba compuesto por más de 30 participantes procedentes de cinco países distintos y de ámbitos diferentes, entre los que se incluían bibliotecas, museos, archivos, organismos oficiales y entidades del sector privado.

Una parte de los objetivos del grupo de trabajo consistía en desarrollar un conjunto fundamental de metadatos de preservación, basados en directrices y recomendaciones de creación, gestión y uso, que pudiesen ponerse en práctica y aplicarse de manera general en una amplia variedad de contextos de preservación digital. Esta tarea del grupo concluyó en mayo del 2005 con la publicación del documento *Data Dictionary for Preservation Metadata: Final Report of the PREMIS working group* (*Diccionario de Datos para Metadatos de Preservación: Informe final del grupo de trabajo PREMIS*).

Este informe de 231 páginas ofrece numerosos recursos sobre metadatos de preservación. El primero y principal es el propio Diccionario de Datos, un material práctico y exhaustivo para la puesta en práctica de metadatos de preservación en sistemas de gestión de objetos digitales. Los metadatos de preservación que define el Diccionario de Datos presentan las siguientes características:

- Contribuyen a la viabilidad, la disponibilidad, la claridad, la autenticidad y la identidad de los objetos digitales en un contexto de preservación.
- Representan la información que la mayoría de los repositorios necesitan conocer para preservar materiales digitales a largo plazo.

- Prestan especial atención a los «metadatos aplicables»: rigurosamente definidos, basados en directrices para su creación, gestión y uso, y orientados a flujos de trabajo automatizados.
- Son técnicamente neutros, es decir, no se presupone el uso de ninguna tecnología de preservación en concreto, ni tampoco de estrategias, sistemas de almacenamiento y gestión de metadatos, etc.

Además del Diccionario de Datos, el grupo de trabajo publicó un esquema XML que permitía incorporar el Diccionario de Datos en sistemas de gestión de objetos digitales. El Diccionario de Datos PREMIS fue premiado con el galardón Digital Preservation Award en el 2005, auspiciado por los British Conservation Awards, y recibió asimismo el Society of American Archivists Preservation Publication Award en el 2006.

Tras la publicación del Diccionario de Datos en el 2005, el grupo de trabajo PREMIS fue disuelto y se puso en marcha la denominada Actividad de Mantenimiento de PREMIS, patrocinada por la Biblioteca del Congreso (*Library of Congress*) de Estados Unidos, con el fin de mantener el Diccionario de Datos y coordinar tanto actividades destinadas a la difusión del conocimiento sobre metadatos de preservación como a otras cuestiones análogas. Además de proporcionar un sitio web permanente para el Diccionario de Datos, un esquema XML y otros materiales relacionados, la Actividad de Mantenimiento también se encarga de gestionar la lista de discusión y la wiki del Grupo de Implementadores de PREMIS, elabora seminarios a cerca de la utilización del Diccionario de Datos y encarga estudios concretos sobre metadatos de preservación. La Actividad de Mantenimiento también creó un Comité Editorial, responsable de ampliar el Diccionario de Datos y el esquema XML, y de promocionar el uso de ambos.

Los miembros del Comité Editorial provienen de diversos países y áreas institucionales.

En el momento de la publicación del Diccionario de Datos, se tomó la decisión de establecer un periodo de carencia de al menos 18 meses, con objeto de conceder a la comunidad de expertos en preservación digital el tiempo suficiente para leer, comprender el material, implementarlo, identificar errores y, principalmente, presentar propuestas de mejora orientadas a incrementar el valor y facilitar el uso del Diccionario de Datos. La recopilación de propuestas se llevó a cabo mediante diversos mecanismos y, en el 2007, el Comité Editorial estimó que se habían

recopilado suficientes comentarios como para que resultase provechoso poner en marcha la primera revisión del Diccionario de Datos. Los miembros del Comité Editorial examinaron el Diccionario e hicieron todo lo posible para involucrar a las partes interesadas en este proceso. El Comité informó en todo momento a la comunidad de expertos en preservación de los temas que se estaban tratando, les pidió que presentaran comentarios a las revisiones y consultó a expertos en otras áreas en aquellos casos en que se consideró oportuno. El resultado de este proceso es el Diccionario de Datos PREMIS de metadatos de preservación, versión 2.0.

1.1.1 Desarrollo del Diccionario de Datos PREMIS original

El grupo de trabajo PREMIS se creó con el fin de retomar la labor emprendida previamente por el grupo de trabajo PMF (*Preservation Metadata Framework*) financiado por OCLC y RLG. En el periodo 2001–2002, el grupo de trabajo PMF definió qué tipo de información debería asociarse a los objetos digitales conservados. Su informe, «*A Metadata Framework to Support the Preservation of Digital Objects*» («Esquema de metadatos empleados en la preservación de objetos digitales», en adelante, el Framework), proponía una lista o prototipo de conjunto de metadatos¹. Sin embargo, para poder implementar el citado prototipo de metadatos era necesario realizar tareas adicionales. Se le encargó al grupo de trabajo PREMIS que trabajara sobre la base conseguida por el grupo PMF y desarrollara un Diccionario de Datos de metadatos fundamentales para sistemas de gestión de objetos digitales y que a su vez definiera y orientara en las mejores prácticas para crear, gestionar y usar metadatos en sistemas de preservación.

Dado que el grupo de trabajo PREMIS se focalizaba más en la vertiente práctica que en la teórica, se buscó a miembros de instituciones que estaban desarrollando repositorios de preservación o trabajaban ya con ellos en el contexto del patrimonio cultural. El grupo de trabajo estaba formado por representantes de bibliotecas académicas y nacionales, museos, archivos, administraciones públicas y empresas privadas de cinco países. Además, PREMIS invitó a un comité consultivo internacional, compuesto por expertos, que se encargaría de evaluar los progresos del proyecto.

Para entender cómo los repositorios de preservación estaban incorporando los metadatos de preservación, en noviembre del 2003, el grupo de trabajo realizó un estudio en unas 70 organizaciones que habían demostrado interés por la preservación digital o ya estaban involucradas en el tema. El estudio brindó la

oportunidad de explorar las últimas novedades relativas a la preservación digital; se formularon preguntas para recabar información sobre políticas, gestión y financiación, arquitectura de sistemas y estrategias de preservación, así como sobre el uso práctico de metadatos. El subgrupo telefoneó a 16 de los 48 encuestados para mantener con ellos entrevistas pormenorizadas. En diciembre del 2004, el grupo de trabajo PREMIS publicó el informe basado en el estudio de repositorios digitales, *Implementing Preservation Repositories for Digital Materials: Current Practice and Emerging Trends in the Cultural Heritage Community* (Incorporación de repositorios de preservación para material digital: prácticas en uso y nuevas tendencias en el mundo del patrimonio cultural, en adelante, el *Implementation Survey Report*²). Los hallazgos de este estudio aportaron información de gran utilidad a los debates del grupo de trabajo mientras éste desarrollaba el Diccionario de Datos.

Tanto el Framework como el Diccionario de Datos PREMIS se adaptan al modelo de referencia Open Archival Information System (OAIS) ISO 14721³, el cual proporciona una base conceptual que adopta la forma de un esquema de clasificación para objetos y paquetes de información además de una estructura para sus metadatos asociados. Si bien el Framework puede ser considerado una aplicación práctica del modelo de referencia OAIS a través del mapeo de los metadatos de preservación a esa estructura conceptual; el Diccionario de Datos PREMIS puede ser entendido como la traducción del Framework a un conjunto de unidades semánticas.

Sin embargo, obsérvese que el Diccionario de Datos y el OAIS en ocasiones difieren en lo que respecta al uso de la terminología; estas diferencias aparecen recogidas en el [glosario](#) que acompaña a este informe. Las diferencias terminológicas a menudo reflejan el hecho de que las unidades semánticas de PREMIS exigen mayor especificidad que las definiciones que proporciona el OAIS, lo que cabe esperar cuando se pasa de un marco conceptual a su implementación.

1.1.2 Metadatos fundamentales de preservación aplicables

El Diccionario de Datos PREMIS define los metadatos de preservación como la información que un repositorio utiliza para llevar a cabo el proceso de preservación digital. En concreto, el grupo se centró en los metadatos destinados al mantenimiento de la viabilidad, la disponibilidad, la claridad, la autenticidad y la

identidad en el contexto de la preservación. Por lo tanto, los metadatos de preservación engloban una serie de categorías de metadatos que normalmente se usan para diferenciarlos: administrativos (incluidos derechos y permisos), técnicos y estructurales. Se prestó especial atención a la documentación sobre la procedencia digital (la historia de los objetos) y a la documentación de las relaciones, especialmente aquellas entre distintos objetos dentro del repositorio de preservación.

El grupo consideró una serie de definiciones para el concepto «fundamental». En una de ellas, fundamental designa todo metadato imprescindible en cualquier circunstancia. En otra, fundamental significa que el metadato es pertinente en todo repositorio que ejecute cualquier tipo de estrategia de preservación. PREMIS opta por la siguiente definición práctica: *aquellos datos que probablemente en la actualidad resulten de utilidad para que casi todos los repositorios puedan llevar a cabo una labor de preservación digital*. El uso de las expresiones probablemente y casi todos en esta definición es intencionado, pues no por considerarse fundamentales deben ser utilizados obligatoriamente en todos los casos: ciertas unidades semánticas se consideran de aplicación optativa.

El concepto de “*aplicabilidad*” también requería una definición. La mayoría de los repositorios de preservación lidian con gran cantidad de datos. Por lo tanto, un factor clave para la aplicabilidad de los metadatos de preservación es la cuestión de que el repositorio pueda generar y procesar valores automáticamente. En los casos en los que fue posible, el grupo definió unidades semánticas que no requerían intervención humana a la hora de su introducción o análisis. Por ejemplo, se prefieren los valores codificados en una lista de autoridades a los textos descriptivos.

El grupo de trabajo decidió que el Diccionario de Datos debía ser totalmente independiente de la forma en que se aplicara. Es decir, los metadatos fundamentales definen la información que un repositorio necesita, y esto es independiente del modo en que se almacena esa información, e incluso del hecho de que pueda no llegar a almacenarse. Por ejemplo, para que un identificador dado se pueda utilizar adecuadamente, lo que es necesario es conocer el esquema de identificadores y el espacio de nombres (namespace) al que pertenece. Si un repositorio concreto utiliza solamente un tipo de identificador, no necesitará registrar el esquema asociado a cada objeto pero sí deberá contar con la información mencionada y ser capaz de proporcionarla a la hora de intercambiar

metadatos con otros repositorios. Dado este énfasis en la necesidad de conocer, más que en la necesidad de registrar o representar formas concretas, el grupo prefirió el uso del término *unidad semántica* en lugar del de metadato. El Diccionario de Datos denomina y describe unidades semánticas.

1.2 El modelo de datos PREMIS

El grupo de trabajo desarrolló un sencillo modelo de datos para organizar las unidades semánticas definidas en el Diccionario de Datos. Este modelo define cinco entidades que el grupo de trabajo consideró especialmente importantes en el contexto de la preservación digital: Entidades Intelectuales, Objetos, Acontecimientos, Derechos y Agentes⁴. Cada unidad semántica definida en el Diccionario de Datos es una propiedad de una de las entidades del modelo de datos. El Gráfico 1 ilustra el modelo de datos PREMIS.

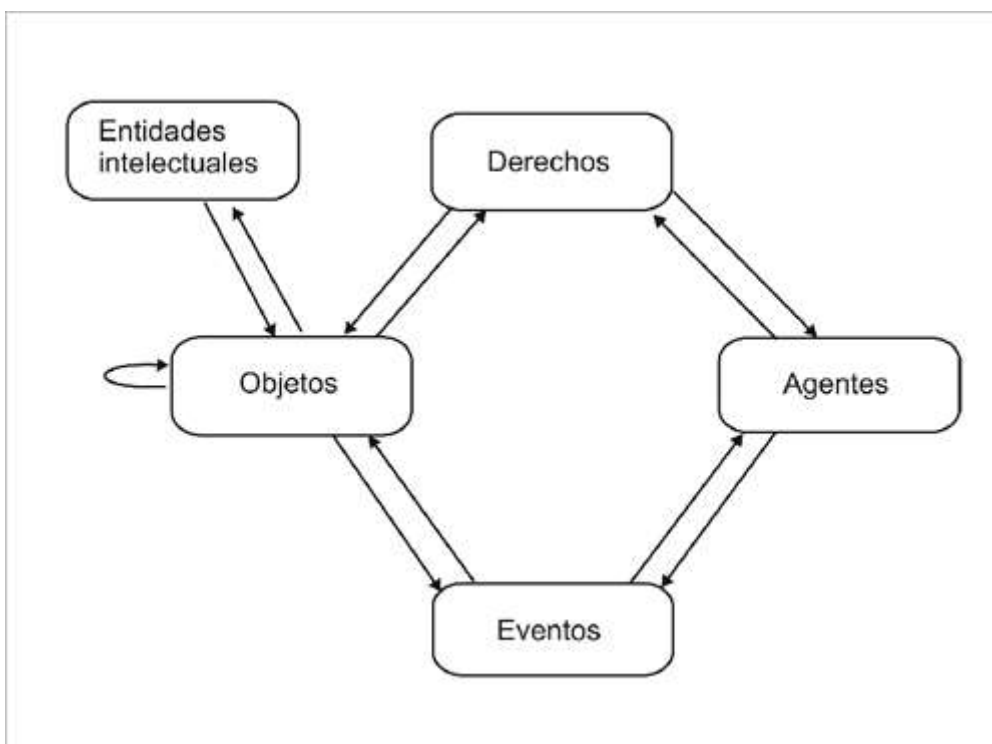


Gráfico 1: El modelo de datos PREMIS

En el **Gráfico 1**, las entidades se representan mediante recuadros y las relaciones entre entidades mediante flechas. La dirección de la flecha indica el sentido de la conexión de la relación, tal y como se registra en los metadatos de preservación. Por ejemplo, la flecha que apunta desde la entidad Derechos hacia la entidad

Agentes indica que el metadato asociado con la entidad Derechos incluye una unidad semántica con información sobre la relación con un Agente.

La flecha que surge de la entidad Objetos y apunta de nuevo a ésta indica que las unidades semánticas definidas en el Diccionario de Datos soportan el registro de las relaciones entre Objetos. Este tipo de relaciones no afecta a ninguna otra entidad del modelo de datos: mientras que los Objetos se pueden relacionar con otros Objetos, los Acontecimientos no se pueden relacionar con otros Acontecimientos, ni los Agentes con otros Agentes, y así sucesivamente.

Las entidades en el modelo de datos PREMIS se definen de la siguiente manera:

Entidad Intelectual: conjunto de contenidos que se considera una única unidad intelectual a efectos de gestión y descripción, por ejemplo, un libro, un mapa, una fotografía o una base de datos.

Una Entidad Intelectual puede comprender otras Entidades Intelectuales. Por ejemplo, un sitio web puede incluir una página web o una página web puede incluir una imagen. Una Entidad Intelectual puede tener una o más representaciones digitales.

Objeto [digital]: unidad discreta de información en formato digital⁵.

Acontecimiento: acción que al menos afecta a un Objeto o Agente asociado o conocido por el repositorio de preservación.

Agente: persona, organización o programa/sistema informático asociado a los Acontecimientos durante la vida de un Objeto, o a los Derechos ligados a un objeto.

Derechos: declaración de uno o varios derechos o permisos pertenecientes a un Objeto o Agente.

El Diccionario de Datos PREMIS define **unidades semánticas**. Cada unidad semántica especificada en el Diccionario de Datos se mapea a una de las entidades del modelo de datos. En este sentido, una unidad semántica puede entenderse como una propiedad de una entidad. Por ejemplo, la unidad semántica *size* es una propiedad de la entidad Objeto. Las unidades semánticas poseen valores: para un objeto concreto el valor de *size* puede ser «843200004».

En la mayoría de los casos, una unidad semántica concreta es, de forma inequívoca, una propiedad de un solo tipo de entidad. El tamaño de un Objeto es claramente una propiedad de la entidad Objeto. Sin embargo, en algunos casos, una unidad semántica corresponde al mismo tiempo a dos o más tipos de entidad. Por ejemplo, los Acontecimientos tienen consecuencias.

Si un acontecimiento de migración crea un archivo que ha perdido alguna característica importante, la pérdida de esa característica puede considerarse una consecuencia del Acontecimiento y, por lo tanto, una propiedad de la entidad Acontecimiento. O bien puede que se considere como un atributo del nuevo archivo y, por lo tanto, una propiedad de la entidad Objeto. Cuando una unidad semántica se aplica a la vez a múltiples tipos de entidad, la unidad se asocia a un único tipo de entidad en el Diccionario de Datos. El modelo de datos se basa en vínculos entre las diferentes entidades para esclarecer estas relaciones. En el ejemplo anterior, la pérdida de la característica se advierte como una consecuencia específica del Acontecimiento, donde el Acontecimiento contiene el identificador del Objeto implicado. Lo importante es que esta asociación es arbitraria y no implica la necesidad de una implementación específica.

En algunos casos, una unidad semántica adquiere la forma de un **contenedor** que agrupa un conjunto de unidades semánticas relacionadas. Por ejemplo, la unidad semántica *identifier* (identificador) agrupa las unidades semánticas *identifierType* (tipo de identificador) e *identifierValue* (valor del identificador). Las subunidades agrupadas se conocen cómo los **componentes semánticos** del contenedor. Algunos contenedores se definen cómo **contenedores de extensión**, que permiten el uso de metadatos codificados según un esquema externo. Esto posibilita la extensión de PREMIS con elementos de metadatos de detalle, es decir, elementos que no se consideran fundamentales o que están fuera del ámbito del Diccionario de Datos.

Una **relación** es una declaración de asociación entre casos individuales de entidades. El término relación puede interpretarse en sentido lato o estricto, y expresarse de maneras muy diferentes. Por ejemplo, la afirmación «el Objeto A tiene un formato B» puede considerarse una relación entre A y B. El modelo PREMIS, sin embargo, trata el formato B como una propiedad del Objeto A. PREMIS reserva el concepto de relación para asociaciones entre dos o más entidades Objeto o entre entidades de distintos tipos, tales como un Objeto y un Agente.

1.2.1 Más información sobre Objetos

La entidad Objeto tiene tres subtipos: fichero, cadena de bits y representación.

Un **fichero** es una secuencia de bytes, con un nombre y orden, y reconocida por un sistema operativo. Un fichero puede contener cero o más bytes y posee un formato y unos permisos de acceso además de las típicas características de sistema como tamaño y fecha de su última modificación.

Una **cadena de bits** consiste en una serie de datos contiguos o no contiguos dentro de un fichero que tiene propiedades comunes en lo que respecta a su preservación. Una cadena de bits no puede convertirse en un fichero autónomo sin que se añada una estructura de fichero (cabecera, etc.) o sin que se modifique su estructura de bits para adaptarlo a un formato de fichero concreto.

Una **representación** es el conjunto de ficheros, incluidos los metadatos estructurales, necesarios para recuperar, en su totalidad y manteniendo el sentido, una Entidad Intelectual. Por ejemplo, un artículo de periódico puede estar compuesto por un único fichero PDF; este fichero por sí solo constituye la representación. Otro artículo de periódico puede constar de un fichero SGML y dos ficheros de imagen; estos tres ficheros constituyen la representación. Un tercer artículo puede quedar representado por una imagen TIFF para cada una de sus doce páginas más un fichero XML de metadatos estructurales en el que se indica el orden de las páginas; estos trece ficheros constituyen la representación.

Ficheros, cadenas de bits y cadenas de ficheros

En el modelo de datos PREMIS, un fichero es similar al concepto tradicional de fichero informático: un conjunto de ceros o más bytes reconocibles por un sistema operativo. Los ficheros se pueden leer, escribir y copiar, y tienen nombres y formatos.

Según se define en el modelo de datos PREMIS, una cadena de bits es un conjunto de bits insertados dentro de un fichero. Éste se diferencia del fichero de uso habitual en que una cadena de bits, en teoría, podría abarcar más de un fichero. Un claro ejemplo de fichero con cadenas de bits embebidas es un fichero TIFF que contenga dos imágenes.

De acuerdo con las especificaciones de formato de fichero TIFF, éste debe contener una cabecera, con determinada información sobre el fichero y, además, una o

varias imágenes. En el modelo de datos PREMIS, cada una de estas imágenes es una cadena de bits y puede contar con propiedades tales como identificadores, localización, inhibidores y metadatos técnicos detallados (por ejemplo, espacio de color).

Unas cadenas de bits pueden tener las mismas propiedades que los ficheros mientras que otras no. La imagen embebida dentro del fichero TIFF posee claramente propiedades distintas a las del propio fichero. Sin embargo, en otro ejemplo, se podrían agrupar tres ficheros TIFF en un fichero TAR de mayor tamaño y en este caso, los tres ficheros TIFF son cadenas de bits embebidas y todos ellos poseen las propiedades de ficheros TIFF.

El modelo de datos PREMIS limita la definición de cadena de bits de modo que sólo pueda constar de una cadena de bits embebida incapaz de convertirse en un fichero autónomo a menos que se le añada estructura de fichero (por ejemplo, cabeceras) o experimente algún otro cambio que le haga cumplir con determinadas especificaciones de formato de fichero.

Algunos ejemplos de estas cadenas de bits serían una imagen dentro de un fichero TIFF 6.0, datos de audio dentro de un fichero WAVE o gráficos dentro de un fichero de Microsoft Word.

Algunas cadenas de bits embebidas pueden convertirse en ficheros autónomos sin que sea necesario añadirles información adicional alguna, aunque puede que haya que realizar algún proceso de transformación, como descompresión, descifre o decodificación sobre la cadena de bits durante el proceso de extracción. Algunos ejemplos de estas cadenas de bits pueden ser un TIFF dentro de un fichero TAR o un EPS codificado dentro de un fichero XML.

En el modelo de datos PREMIS, estas cadenas de bits se definen como cadenas de ficheros, es decir, ficheros reales embebidos en ficheros mayores. Las cadenas de ficheros poseen todas las propiedades de los ficheros, no así las cadenas de bits. En el Diccionario de Datos, la columna «Fichero» incluye tanto los ficheros como las cadenas de ficheros. La columna «Cadena de bits» se refiere al subconjunto de cadenas de bits que no son cadenas de ficheros y que se ciñen estrictamente a la definición de cadena de bits de PREMIS. La localización (*contentLocation* en el Diccionario de Datos) de un fichero, por lo general, corresponde a una ubicación en el almacenamiento, mientras que la localización de una cadena de ficheros o de bits

se correspondería con el punto de inicio (*offset*) dentro del fichero del que forma parte.

Representaciones

A menudo, el objetivo de los repositorios de preservación es permitir la utilización de las entidades intelectuales a largo plazo. Para que una entidad intelectual pueda mostrarse, reproducirse o ser utilizada por una persona, todos los ficheros que conforman, al menos una versión de esa entidad intelectual, deben estar identificados, almacenados y conservados de tal forma que se puedan reunir y recuperar para el usuario en todo momento. El conjunto de ficheros necesarios para ello es lo que conocemos como representación.

PREMIS escogió este término para evitar el vocablo manifestación, ya que éste se emplea en los *Functional Requirements for Bibliographic Records* (requisitos funcionales de los registros bibliográficos, FRBR)⁶. En los FRBR, una entidad Manifestación incluye «todos los objetos físicos que poseen las mismas características en cuanto a contenido intelectual y forma física». En el modelo PREMIS, una representación *es la única instancia en formato digital de una entidad intelectual contenida en un repositorio de preservación*.

Un repositorio de preservación puede albergar más de una representación para la misma entidad. Por ejemplo, el repositorio puede albergar una única imagen (una «estatua de un caballo») como fichero TIFF. En algún momento, el repositorio crea un fichero derivado JPEG2000 a partir del TIFF y guarda ambos ficheros. Cada uno de estos ficheros constituiría una representación de «estatua de un caballo».

En un ejemplo más complejo, «estatua de un caballo» puede ser una parte de un artículo formado por esa imagen TIFF y un fichero de texto SGML codificado. Si el repositorio hubiera creado una versión JPEG2000 del TIFF, guardaría dos representaciones del artículo: el fichero TIFF y el SGML constituirían una representación, mientras que el fichero JPEG2000 y el SGML constituirían otra.

La manera en que se almacenan esas representaciones depende de cada implementación. Un repositorio puede decidir almacenar una única copia del fichero SGML, que después compartirían las distintas representaciones. Otra posibilidad

consistiría en que el repositorio optara por duplicar el fichero SGML y almacenar dos copias idénticas. En este caso, las dos representaciones serían la copia 1 del TIFF y el SGML y la copia 2 del JPEG2000 y el SGML.

No todos los repositorios de preservación manejan representaciones. Por ejemplo, puede que un repositorio únicamente conserve objetos fichero y deje en manos de agentes externos la creación de representaciones a partir de estos objetos. Si el repositorio no gestiona representaciones, tampoco necesita almacenar metadatos sobre ellas.

1.2.2 Entidades Intelectuales y Objetos

La relación entre Entidades Intelectuales y Objetos puede ilustrarse con un par de ejemplos:

Ejemplo 1, *Animal Antics*: el libro *Animal Antics* se publicó en 1902. Una biblioteca lo digitalizó y creó un fichero TIFF para cada una de las 189 páginas. Como metadato estructural creó un fichero XML con la información necesaria para recomponer el libro completo con las imágenes. A continuación, la biblioteca aplicó un programa de OCR (reconocimiento óptico de caracteres) sobre las imágenes TIFF para generar un único fichero de texto que se etiquetó a mano en SGML. La biblioteca entregó 189 ficheros TIFF, un fichero XML y un fichero SGML a un repositorio de preservación.

Para el repositorio, *Animal Antics* es una Entidad Intelectual: se trata de una unidad razonable que puede describirse como un todo, con propiedades tales como autor, título y fecha de publicación. El repositorio dispone de dos representaciones: una que consta de 189 ficheros TIFF y un fichero XML, y otra que consta de un fichero SGML. Cada representación podría recuperar una versión completa de *Animal Antics*, aunque con distintas funcionalidades. El repositorio registrará metadatos sobre dos objetos de representación y 191 objetos fichero.

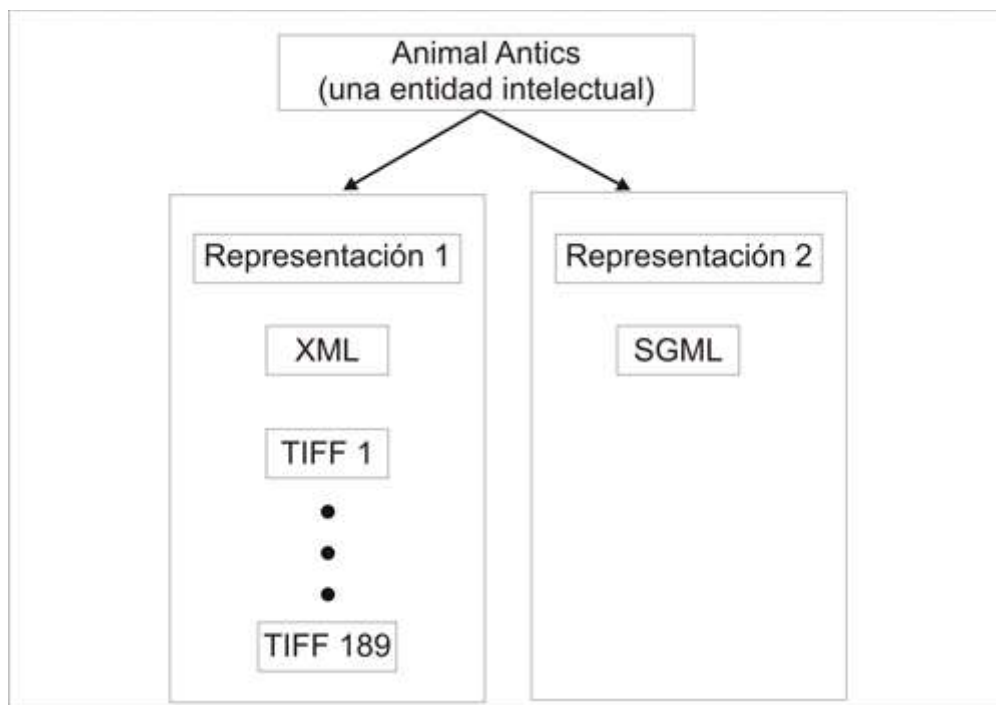


Gráfico 2: Ejemplo de la Entidad Intelectual Animal Antics

Ejemplo 2, *Welcome to U*: *Welcome to U*, depositado en un repositorio de preservación bajo la forma de un fichero AVI (Audio-Video Interleaved), es una película de 10 minutos que da la bienvenida a un campus universitario a los nuevos estudiantes.

Welcome to U es una Entidad Intelectual. El repositorio tiene una representación, que consta de un único fichero AVI. La estrategia de preservación del repositorio requiere que los bits de audio del fichero AVI se traten con independencia de los bits de vídeo. El repositorio registrará metadatos sobre un objeto de representación, un objeto fichero y dos objetos de cadenas de bits.

Más información sobre Acontecimientos

La entidad Acontecimiento agrupa metadatos con información sobre acciones. Hay numerosas razones que motivan a un repositorio de preservación a registrar acontecimientos. La documentación de acciones que modifican un objeto digital (es decir, aquellas que crean una nueva versión de éste) resulta crucial para el mantenimiento de la procedencia digital, un elemento clave para demostrar su autenticidad. Las acciones que generan nuevas relaciones o alteran las existentes son importantes a la hora de explicar esas relaciones. Incluso, el registro de las acciones que no entrañan ninguna alteración, tales como las comprobaciones de

validez e integridad de objetos, pueden resultar importantes para la gestión. Algunos repositorios pueden registrar acciones como por ejemplo solicitudes de acceso o informes con vistas a suministrar información.

Es potestad del repositorio decidir qué acciones quedarán registradas como Acontecimientos. Algunas acciones pueden considerarse demasiado triviales como para recopilarlas o bien pueden quedar registradas en otros sistemas (por ejemplo, las copias de seguridad rutinarias de los ficheros pueden recopilarse en sistemas de gestión de almacenamiento). También requiere una toma de decisión la idea de registrar o no Acontecimientos que suceden antes de que un objeto entre a formar parte de un repositorio de preservación por ejemplo, la derivación a partir de otro objeto o los cambios de custodia. En teoría, los Acontecimientos que ocurran con posteridad al expurgo de una Entidad Intelectual también se podrían registrar. Por ejemplo, un repositorio puede dar de baja primero una Entidad Intelectual, suprimir después todos los objetos fichero asociados a ella y registrar cada supresión como un Acontecimiento.

En el modelo de datos, los Objetos se relacionan con los Acontecimientos de dos formas. Si un Objeto se relaciona con otro a causa de un Acontecimiento el identificador del Acontecimiento se registra tanto en el contenedor *relationship* como en el componente semántico *relatedEventIdentification*. Si el Objeto simplemente tiene asociado un Acontecimiento sin ninguna relación con un segundo Objeto, el identificador del Acontecimiento se registra en el contenedor *linkingEventIdentifier* (identificador del acontecimiento vinculado). Para obtener información adicional sobre relaciones, véase [relaciones entre objetos](#).

Por ejemplo, supongamos que se incorpora un fichero XML (objeto A) en un repositorio de preservación y durante dicho proceso se crea una versión normalizada de éste (Objeto B) mediante la ejecución de un programa (Acontecimiento 1). En los metadatos para el Objeto B las relaciones podrían quedar registradas de la siguiente manera:

```
relationshipType = "derivation"
```

```
relationshipSubType = "derived from"
```

```
relatedObjectIdentification
```

```
relatedObjectIdentifierType = "local"
```

```
relatedObjectIdentifierValue = "A"
```

relatedObjectSequence = "not applicable"

relatedEventIdentification

relatedEventIdentifierType = "local"

relatedEventIdentifierValue = "1"

relatedEventSequence = "not applicable"

Siguiendo con este ejemplo, supongamos que después de la creación del Objeto B, éste se valida mediante la ejecución de otro programa (Acontecimiento 2). En este caso, el Acontecimiento 2 sólo es relevante para el Objeto B, y no para la relación existente entre B y A. El vínculo con el acontecimiento 2 se registraría como *linkingEventIdentifier*:

linkingEventIdentifierType = "local"

linkingEventIdentifierValue = "2"

Un Objeto dado puede asociarse de estas dos formas con cualquier número de Acontecimientos.

Todos los Acontecimientos tienen consecuencias (aciertos, errores, etc.). Algunos Acontecimientos también generan productos; por ejemplo, la ejecución de un programa produce un nuevo Objeto fichero. Las unidades semánticas *eventOutcome* (consecuencias del acontecimiento) y *eventOutcomeDetail* (detalles sobre las consecuencias del acontecimiento) se utilizan para documentar consecuencias. Por ejemplo, si el acontecimiento es una acción de validación de formato, el valor de *eventOutcome* (consecuencias del acontecimiento) puede ser un código que indique que el objeto es completamente válido o un código que indique que no lo es. En este último caso, *eventOutcomeDetail* se puede usar para describir las anomalías que se encuentren. Si el programa que lleva a cabo la validación crea un registro de avisos y mensajes de error, un segundo *eventOutcomeDetail* se podría utilizar para almacenar o señalar dicho valor.

Si un acontecimiento crea objetos que están almacenados en el repositorio, estos objetos deben describirse como entidades con un conjunto completo de metadatos pertinentes y asociados al acontecimiento mediante enlaces.

1.2.3 Más información sobre Agentes

Los Agentes, sin dejar de tener su importancia, no constituyen el centro de atención del Diccionario de Datos, el cual únicamente define la forma de identificar el agente y la clasificación del tipo de agente (persona, organización o software).

Si bien pueden ser necesarios más metadatos, se proporciona la posibilidad de recurrir a iniciativas externas para definir esta área.

El diagrama de modelo de datos muestra una flecha desde la entidad Agente hasta la entidad Acontecimiento, pero no hay ninguna flecha desde el Agente hasta la entidad Objeto. Los Agentes ejercen sólo una influencia indirecta sobre los Objetos a través de los Acontecimientos. Cada Acontecimiento puede tener uno o más Objetos relacionados y uno o más Agentes relacionados. Debido a que un solo Agente puede llevar a cabo diferentes funciones en diferentes Acontecimientos, la función del Agente es una propiedad de la entidad Acontecimiento y no de la entidad Agente.

1.2.4 Más información sobre Derechos

Se han dedicado muchos esfuerzos al estudio de los metadatos relativos a derechos y permisos de propiedad intelectual, desde la terminología empleada hasta el modelo <indecs>. Sin embargo, han sido escasos los recursos dedicados a los permisos y derechos relacionados específicamente con la preservación digital. Tras la publicación de la primera edición del Diccionario de Datos PREMIS, la Biblioteca del Congreso de Estados Unidos, organismo competente para el mantenimiento de PREMIS, encomendó a Karen Coyle⁷ la redacción del documento «Rights in the PREMIS Data Model» («Los derechos de autor en el modelo de datos PREMIS»). Este documento abordaba los derechos de autor, las licencias y la legislación como si fueran la base para el establecimiento de los derechos de propiedad intelectual y recomendaba ampliar el apartado de derechos de autor del Diccionario de Datos con el fin de incluir información sobre dichos fundamentos.

Como consecuencia, la *permissionStatement* (mención de permiso) en el Diccionario de Datos original fue reemplazada por la *rightsStatement* (mención de derecho) en esta versión. El Comité Editorial tuvo especialmente en cuenta en esta revisión el artículo de Coyle, otros materiales de referencia como el excelente «Digital Preservation and Copyright» («El copyright y la preservación digital»)⁸ de Peter Hirtle y el borrador del módulo del copyrightMD de la Biblioteca Digital de California (California Digital Library)⁹. Cabe señalar que los usos propuestos para

copyrightMD y para los derechos de PREMIS son bastante diferentes. El esquema del copyrightMD aspira a proporcionar una documentación objetiva que permita a las personas realizar valoraciones fundamentadas sobre los derechos de autor de un trabajo determinado. La *rightsStatement* (mención de derechos) de PREMIS se definió con el fin de permitir que un repositorio de preservación estableciera si tiene derecho o no para llevar a cabo una determinada acción de forma automatizada y en tal caso con cierta documentación que avale dicha decisión.

1.3 Cuestiones generales sobre la estructura y el uso del Diccionario de Datos

Las unidades semánticas que se definen en el Diccionario de Datos PREMIS están vinculadas entre sí por algunas convenciones estructurales que ayudan a organizar el Diccionario y a llevar a cabo su implementación.

Estas convenciones incluyen el uso de identificadores, la manera en la que se gestionan las relaciones en el Diccionario de Datos y el «1:1 Principle» (principio 1:1) que relaciona metadatos con Objetos.

1.3.1 Identificadores

Los casos relativos a Objetos, Acontecimientos, Agentes y Derechos están identificados de manera inequívoca por un conjunto de unidades semánticas recogidas en contenedores “*identificadores*”. Estas unidades semánticas siguen una sintaxis y una estructura idénticas independientemente del tipo de entidad:

[entity type]Identifier

[entity type]IdentifierType: dominio en el que el identificador es único

[entity type]IdentifierValue: conjunto de caracteres del identificador

Los ejemplos siguientes ilustran el uso de esta sintaxis para identificar un objeto localizado en el Digital Repository Service (DRS) de Harvard y un acontecimiento que tiene lugar bajo los auspicios del Name Resolution Service (NRS):

Ejemplo 1: Identificar un objeto

ObjectIdentifier

ObjectIdentifierType: NRS

ObjectIdentifierValue: <http://nrs.harvard.edu/urn-3:FHCL.Loeb:sa1>

Ejemplo 2: Identificar un acontecimiento

EventIdentifier

EventIdentifierType: NRS

EventIdentifierValue: 716593

En los dos ejemplos, el tipo de identificador es «NRS», lo que indica que el identificador es único dentro del dominio del Name Resolution Service que asigna identificadores al Digital Repository Service. El tipo de identificador debe definirse de la manera más específica posible y ofrecer cuanta información sea precisa para indicar qué autoridad tiene la competencia de dar el nombre y cómo construir el valor del identificador. Por ejemplo, habría sido aceptable usar «URL» para ObjectIdentifierType en el primer ejemplo, ya que el valor del identificador es único en ese dominio, pero «NRS» transmite más información acerca del dominio en el que se crea y utiliza el identificador.

Si todos son identificadores locales del repositorio, es muy poco probable que sea necesario registrar explícitamente cada tipo de identificador del sistema. Este es un ejemplo de unidad semántica cuya información se conoce implícitamente por el contexto o la política y, por lo tanto, no se incorpora como un elemento de metadato en el sistema de preservación. Sin embargo, si el repositorio intercambiara tanto objetos digitales como sus metadatos asociados con otros repositorios, el tipo de identificador debería indicarse explícitamente.

Los identificadores pueden crearse dentro o fuera del repositorio. El Diccionario de Datos PREMIS no exige, ni siquiera recomienda, un esquema de identificadores en concreto. Esta cuestión depende exclusivamente de cada implementación y, por lo tanto, se encuentra fuera del ámbito del Diccionario de Datos, el cual se limita a proporcionar una sintaxis genérica que se puede utilizar para expresar el tipo y el valor del identificador, sea cual sea el esquema específico elegido. Sin embargo, es recomendable, siempre que sea posible, que los repositorios elijan esquemas de identificación persistentes.

Los identificadores son repetibles para Objetos y Agentes; pero *no pueden ser repetibles* para Derechos y Acontecimientos.

Los Objetos y los Agentes a menudo tienen múltiples identidades en un entorno global y en distintos sistemas y, por lo tanto, es probable que tengan múltiples

identificadores. Se considera que los Derechos y los Acontecimientos están limitados al contexto específico de un repositorio de preservación concreto y, por consiguiente, no requieren múltiples identificadores.

Los identificadores se utilizan como referencias para establecer relaciones entre entidades del modelo de datos PREMIS. Estas relaciones se abordan en la sección siguiente.

1.3.2 Relaciones entre Objetos

Tal y como se explica anteriormente, un Objeto de un repositorio se puede relacionar con uno o más Objetos del citado repositorio. El Diccionario de Datos PREMIS proporciona unidades semánticas para ilustrar las relaciones entre Objetos. El grupo de trabajo comenzó la investigación de esta cuestión recopilando ejemplos de proyectos de metadatos de preservación. Se encontró una gran variedad de metadatos expresados como relaciones —por ejemplo, “is migrated from,” “is keyed text of,” “is thumbnail of.”. En algunos casos, estas relaciones combinan más de un hecho (por ejemplo, “is keyed text of” combina con “is a keyed text” y “is derived from”. El grupo también examinó cómo refinar elementos del Dublin Core Relation: IsPartOf, IsFormatOf, IsVersionOf, etc. y concluyó que la mayoría de las relaciones entre objetos parecen ser variantes de la siguiente tipología básica: estructural, de derivación y de dependencia.

Las relaciones estructurales muestran relaciones entre las partes de los objetos. Las relaciones estructurales entre los ficheros que constituyen una representación de una Entidad Intelectual son claramente metadatos de preservación fundamentales. Si un repositorio de preservación no es capaz de volver a ordenar las piezas de un objeto digital, esto significa que no ha conseguido preservar el objeto. En un objeto digital simple (por ejemplo, una fotografía), la información estructural es mínima: el fichero constituye la representación. Sin embargo, otros objetos digitales, como los libros electrónicos y los sitios web, pueden tener relaciones estructurales muy complejas.

Las relaciones de derivación son el resultado de la duplicación o transformación de un Objeto. El contenido intelectual del Objeto resultante es el mismo, pero la representación del Objeto —y, posiblemente, su formato— son diferentes. Cuando se migra el fichero A con formato X para crear el fichero B con formato Y, existe una relación de derivación entre A y B.

Muchos objetos digitales son complejos y tanto la información estructural como la de derivación pueden cambiar como resultado de las actividades de preservación. Por ejemplo, un libro digitalizado representado por 400 páginas de imágenes TIFF podría convertirse, tras la migración, en cuatro ficheros PDF de 100 páginas cada uno.

Se puede establecer una relación estructural entre objetos mediante un acto de derivación antes de su inclusión en el repositorio. Por ejemplo, un documento procedente de un procesador de texto podría haber sido la base para crear ficheros derivados en formato PDF y XML. Si sólo se introducen los ficheros PDF y XML en el repositorio de preservación, estos objetos serán representaciones diferentes de la misma Entidad Intelectual y mantendrán una relación «padre-hijo» con el fichero fuente del procesador de texto. No guardarán relación de derivación con ningún otro, pero sí una relación estructural de «hermanos» (hijos de un mismo padre).

No existe una única manera de dar forma a toda la información estructural o de derivación, por lo que, en lugar de ceñirse a un determinado enfoque, el grupo decidió simplemente identificar la información esencial. El Diccionario de Datos PREMIS la describe en los componentes semánticos de la relación de unidades semánticas. Las relaciones estructurales y de derivación enlazan Objetos y éstos deben identificarse.

El tipo de relación se debe identificar de alguna manera (por ejemplo, "is child of") y la relación puede asociarse con el Acontecimiento que la creó. Las personas encargadas de la implementación posiblemente elegirán la forma que mejor se adapte al contenido que se preserva usando, por ejemplo, el METS¹⁰ structMap o los esquemas de metadatos descriptivos que definen los tipos de relaciones (por ejemplo, Dublin Core¹¹).

Se establece una **relación de dependencia** cuando un objeto necesita a otro para realizar su función, representarse o dar coherencia a su contenido. Un objeto puede requerir un tipo de letra, una hoja de estilo, una DTD (Definición de Tipo de Documento), un esquema u otro fichero que no sea parte formal del objeto en sí pero que sea necesario para recuperarlo. El Diccionario de Datos maneja relaciones de dependencia como parte de la información del entorno, en *dependency* y en *swDependency* de las unidades semánticas. De este modo, los requisitos de hardware y software y de ficheros dependientes son elementos necesarios para

terminar de comprender qué es lo que se precisa para recuperar o entender un objeto.

1.3.3 Relaciones entre entidades de diferentes tipos

El diagrama de modelo de datos utiliza flechas para mostrar las relaciones entre entidades de tipo diferente.

Los Objetos están relacionados con las Entidades Intelectuales y con los Acontecimientos, los Agentes están relacionados con los Acontecimientos, etc. El Diccionario de Datos expresa las relaciones como información enlazada, incluyendo en la información de la entidad A un puntero a la entidad B relacionada. Cada entidad tiene en el modelo de datos un único identificador que se utiliza como puntero. Así, por ejemplo, la entidad Objeto tiene una flecha que señala a las Entidades Intelectuales y los Acontecimientos. En el Diccionario de Datos, éstos se incorporan mediante las unidades semánticas *linkingIntellectualEntityIdentifier*) y *linkingEventIdentifier*.

1.3.4 El principio 1:1

En preservación digital es una práctica común crear nuevas copias o versiones de los objetos almacenados. Por ejemplo, en las migraciones, un programa puede transformar el fichero A con formato X en el fichero B en formato Y. Hay dos formas de referirse a los ficheros A y B. Se pueden entender como un Objeto simple, cuya historia incluye la transformación de X a Y, o bien como dos objetos distintos con una relación creada por el Acontecimiento de transformación.

Con respecto a los metadatos, por el principio 1:1 se considera que cada descripción hace referencia a un recurso exclusivamente.

En el caso de los metadatos de PREMIS, cada Objeto contenido en el repositorio de preservación (fichero, cadena de bits, representación) se describe como un conjunto de bits estático. No es posible cambiar un fichero (ni una cadena de bits ni una representación); únicamente se puede crear uno nuevo (o una cadena de bits o una representación) que esté relacionado con el Objeto fuente. En el ejemplo anterior, por lo tanto, los ficheros A y B son dos objetos distintos pero con una relación derivada entre ellos. El Diccionario de Datos cuenta con una unidad semántica para la fecha de creación de un Objeto *dateCreatedByApplication* pero no para la fecha de modificación del Objeto, pues, por definición, un objeto no se puede modificar.

Cuando se derivan Objetos de los ya existentes, el Acontecimiento que ha creado el nuevo Objeto debe registrarse como un Acontecimiento, que tendrá la fecha y la hora registradas. La relación o relaciones entre los Objetos deberán registrarse utilizando la unidad semántica *relationship* (relación) asociada a la entidad Objeto.

El componente semántico *relatedEventIdentification* (identificación del acontecimiento relacionado) deberá usarse para crear la asociación con el Acontecimiento.

1.4 Consideraciones relativas a la implementación

1.4.1 Conformidad con PREMIS

Para que exista conformidad con PREMIS, el repositorio de preservación debe respetar las especificaciones señaladas en el Diccionario de Datos. Por ejemplo, si el repositorio que dice ser conforme al modelo PREMIS utiliza un metadato con el mismo nombre que el de una unidad semántica del Diccionario de Datos, se espera que este metadato del repositorio comparta igualmente la misma definición. También pueden utilizarse metadatos no definidos en el Diccionario de Datos, pero estos elementos ajenos a PREMIS no deberían interferir ni solaparse con unidades semánticas PREMIS que tengan el mismo nombre. Asimismo se deben respetar las restricciones de datos y las pautas de aplicabilidad del Diccionario de Datos. Para las posibilidades de repetición y la obligatoriedad, PREMIS permite una implementación más restrictiva pero no más liberal. Es decir, una unidad semántica definida en el Diccionario de Datos como repetible puede ser tratada como no repetible en un repositorio, pero no al revés.

El Diccionario de Datos PREMIS designa como obligatorias algunas unidades semánticas cuando describen representaciones, ficheros o cadenas de bits. Las unidades semánticas obligatorias representan la mínima cantidad de información necesaria, en primer lugar, para asegurar la preservación de los objetos digitales a largo plazo y, en segundo lugar, para acompañar a un objeto digital cuando se transfiere desde un repositorio de preservación a otro. No hay una estrategia establecida para recoger, almacenar o manejar las unidades semánticas obligatorias dentro de los sistemas internos del repositorio. Tampoco hay un nivel mínimo de información que deba quedar registrada y conservada explícitamente en el repositorio. En general, las unidades semánticas obligatorias del Diccionario de

Datos representan la información que un repositorio de preservación debe ser capaz de asociar a todo objeto digital archivado.

Los significados específicos de asociación (por ejemplo, almacenamiento de metadatos locales, registros compartidos, etc.) son aspectos relativos a su implementación y se encuentran fuera del ámbito del Diccionario de Datos.

Cuando un objeto digital es intercambiado entre dos repositorios de preservación, el repositorio que remite el objeto debe ser capaz de extraer, a partir de su propio sistema o bien de otras fuentes externas, la información que necesita para alimentar las unidades semánticas marcadas como obligatorias en el Diccionario de Datos. Esta información ha de coincidir con las especificaciones del Diccionario de Datos y debe acompañar al objeto digital antes de que este se transfiera al segundo repositorio. El grupo de trabajo de PREMIS considera que esta información es la indispensable para que el segundo repositorio acepte la custodia del objeto digital y asuma la responsabilidad de su preservación a largo plazo.

Algunas unidades semánticas de PREMIS equivalen a los metadatos definidos en otros esquemas. Si se recurre a estos últimos para alimentar las unidades semánticas de PREMIS, debe quedar garantizado que esta información coincida con los requisitos y restricciones asociadas a la unidad semántica correspondiente del Diccionario de Datos PREMIS. Armonizar el Diccionario de Datos PREMIS con otros esquemas de metadatos en los casos en los que se solapen ayudaría a minimizar los problemas de conformidad. Por ejemplo, el estándar de metadatos Z39.87 (*Technical Metadata for Digital Still Images*, metadatos técnicos para imágenes digitales fijas)¹² revisó algunos de sus elementos con el fin de armonizarlos a las unidades semánticas equivalentes del Diccionario de Datos PREMIS.

En ocasiones, un repositorio de preservación intercambia objetos digitales con agentes que no son repositorios de preservación propiamente dichos. Cuando uno de ellos envía un objeto a un repositorio de preservación para custodiar sus archivos es poco probable que el remitente pueda proporcionar la totalidad de la información necesaria para cumplimentar las unidades semánticas obligatorias. No obstante, proporcionará un subconjunto de esta información, cuya extensión convendría que estuviese acordada previamente entre el remitente y el repositorio. Sea cual sea la extensión de este subconjunto, toda información proporcionada por el remitente debería ajustarse al Diccionario de Datos. El proceso de carga de objetos (ingest) del repositorio proporcionaría el resto de la información de las unidades semánticas.

Cuando un repositorio difunde un objeto digital archivado, es poco probable que el usuario tenga interés por la totalidad de las unidades semánticas obligatorias asociadas al objeto archivado. En su lugar, el usuario recibirá un subconjunto de estas unidades semánticas. Tal y como sucede en la remisión, sea cual sea el tamaño de este subconjunto, toda la información proporcionada por el repositorio debe ser conforme al Diccionario de Datos.

Alcanzar la interoperabilidad a través de una red de repositorios de preservación y otras partes interesadas requiere un consenso en los metadatos necesarios para la preservación a largo plazo bajo el paraguas de un esquema que se implementará. La búsqueda de la conformidad con PREMIS y la obligatoriedad de las unidades semánticas obligatorias pretenden solventar esta necesidad.

1.4.2 Implementación del modelo de datos

El modelo de datos PREMIS pretende aclarar el significado y el uso de las unidades semánticas del Diccionario de Datos. Su finalidad no es otra que la de establecer una arquitectura para su implementación.

El grupo de trabajo entendía que la mayoría de los repositorios de preservación necesitan manejar en algún grado las entidades conceptuales, Objetos, Agentes, Acontecimientos y Derechos, y consideró útil diferenciar las propiedades de las subclases de objetos, como ficheros, cadenas de ficheros, cadenas de bits y representaciones. Su implementación en un repositorio concreto puede necesitar otro grado de especificidad o incluso definir diferentes categorías de entidades. PREMIS recomienda que cada modelo de datos utilizado esté claramente definido y documentado, y que las decisiones relativas a los metadatos sean coherentes con el modelo.

Se pueden agrupar conjuntos de unidades semánticas y relacionarlos indirectamente con entidades particulares. Por ejemplo, entorno es una propiedad de los objetos. Lógicamente, cada fichero tiene asociados uno o más entornos. Sin embargo, en muchos casos el entorno se determina por el formato del fichero; es decir, todos los ficheros con un formato concreto tendrán el mismo entorno de información. Esto se podría gestionar de diferentes maneras mediante distintas aplicaciones prácticas. Por ejemplo:

- El repositorio 1 utiliza un sistema de bases de datos relacional. Tiene una tabla «fichero» con una fila para cada objeto fichero, y una tabla «entorno» con una fila para cada conjunto único de información del

entorno. La tabla «fichero» se puede unir a la tabla «entorno» para conseguir la información de entorno apropiada para cada fichero.

- El repositorio 2 utiliza un registro externo para obtener información del entorno. Conserva un inventario interno de formatos de fichero y sus claves de acceso para el registro externo. La información del entorno es accesible mediante una interfaz de servicios web que se utiliza para el registro externo y se obtiene dinámicamente cuando se necesita.
- El repositorio 3 utiliza un sistema que muestra las representaciones como contenedores y los ficheros como objetos dentro de esos contenedores. Cada objeto se compone de un par de valores: propiedad/tipo.

Las propiedades definen las funciones de los valores. Las propias descripciones de tipo y de propiedad son objetos cuyos identificadores se obtienen del mismo espacio de nombres que los demás identificadores de objeto. Un objeto fichero puede incluir una propiedad de formato. Como la descripción de formato es también un objeto, puede incluir una propiedad de entorno que, a su vez, puede señalar a un objeto de descripción de entorno. Otra posibilidad es que un objeto fichero incluya directamente una propiedad de entorno.

1.4.3 Almacenamiento de metadatos

El estudio realizado por el Implementation Strategies Subgroup indica cómo los repositorios utilizan arquitecturas diferentes para almacenar metadatos. Lo más habitual es almacenar los metadatos en tablas de bases de datos relacionales. También es frecuente almacenar metadatos como documentos XML en una base de datos XML o como documentos XML conservados junto al contenido de los ficheros de datos. Otros métodos pueden consistir en ficheros planos de formato concreto y bases de datos orientadas a objetos. La mayoría de los encuestados utiliza dos o más de estos procedimientos. Para obtener información adicional, véase el *Implementation Survey Report*².

Entre las ventajas de almacenar los metadatos utilizando bases de datos destacan el acceso rápido y la facilidad de actualización y de uso para realizar consultas e informes. El hecho de almacenar los registros de metadatos como objetos digitales en un repositorio junto con los objetos digitales que describen los metadatos también presenta ventajas: es más difícil separar los metadatos del contenido y pueden aplicarse a los metadatos las mismas estrategias de preservación que se

aplican al contenido. La práctica recomendada consiste en almacenar los metadatos más críticos siguiendo ambos procedimientos.

Los objetos compuestos requieren metadatos estructurales para describir su estructura interna y las relaciones entre sus partes. En el Diccionario de Datos PREMIS, las unidades semánticas que empiezan por «related» (relacionada) y «linking» (vinculada) pueden utilizarse para expresar cierta información estructural simple, aunque sólo en algunos casos esta fórmula será adecuada para el uso del objeto. A menudo, la presentación, la navegación o el procesamiento de un objeto requieren metadatos estructurales más ricos registrados según algún otro estándar, como METS¹⁰, MPEG-21¹³ o SMIL¹⁴. En este caso, el propio fichero que contiene los metadatos estructurales sería un objeto fichero que habría que conservar. Exista o no un fichero de metadatos estructurales independientes como parte de la representación, cuando se exporta una representación a otro repositorio, se deberán proporcionar tanto los metadatos que relacionan los ficheros como los de las representaciones.

1.4.4 Provisión de los valores de los metadatos

La mayoría de los repositorios cuentan con un elevado volumen de materiales, por lo que es preferible automatizar al máximo la creación y el uso de los metadatos. Los valores de las numerosas unidades semánticas que componen PREMIS pueden obtenerse automáticamente, bien programando el análisis de los ficheros bien durante el proceso de carga de objetos (ingesta). En los casos en los que la intervención humana resulte inevitable, el grupo opta por emparejar una unidad semántica que requiere un valor codificado con una segunda unidad semántica que permita una explicación textual.

Cuando la información la proporciona el individuo o la organización que envía los objetos al repositorio, se recomienda que éste intente verificar, siempre que sea posible, dicha información mediante un programa. Por ejemplo, si el nombre de un fichero incluye el tipo de extensión, el repositorio no debería dar por supuesto que la extensión del fichero indica necesariamente el formato, sino que debería tratar de verificar el formato del fichero antes de registrarlo como metadato.

Para facilitar el proceso automático, se recomienda el uso de vocabularios controlados para ciertas unidades semánticas de PREMIS. PREMIS asume que cada repositorio adoptará o definirá el vocabulario controlado que le resulte más oportuno. El Diccionario de Datos indica en qué casos se recomienda el uso de un

vocabulario controlado. No impone ningún vocabulario controlado específico, aunque en algunos casos sugiere valores.

El Comité Editorial de PREMIS llegó a la conclusión de que los responsables de su implementación deberían tener la posibilidad de elegir el vocabulario que desean utilizar y de especificar qué vocabulario ha sido el empleado en cada caso. La decisión de comprobar el uso correcto de los valores del diccionario y el modo de llevarlo a cabo son cuestiones que dependen de la implementación. Con la versión 2.0 del Diccionario de Datos PREMIS, la Actividad de Mantenimiento de PREMIS de la Biblioteca del Congreso de Estados Unidos (*PREMIS Maintenance Activity at the Library of Congress*) está tratando de establecer un mecanismo para registrar y publicar aquellos vocabularios controlados que estén siendo utilizados en las unidades semánticas de PREMIS de tal manera que queden recogidos en el esquema de PREMIS. Los repositorios pueden utilizar estos o bien definir los suyos propios, pero en cualquier caso debería quedar clara la fuente de cada vocabulario controlado a la hora exportar e intercambiar metadatos. Además, la interoperabilidad aumenta si se utilizan y definen vocabularios comunes.

El implementador puede elegir si documenta los vocabularios controlados utilizados en su repositorio de tal modo que los compañeros con los que los intercambia puedan prever los valores de los metadatos.

Por ejemplo, aquellos que utilicen METS¹⁰ pueden especificar los vocabularios controlados utilizados en los metadatos en el perfil METS, o bien pueden establecer los perfiles PREMIS para documentar dichos vocabularios. Los esquemas PREMIS XML permiten registrar la fuente. Otros tipos de XML permiten desarrollar mecanismos para identificar vocabularios controlados en uso o para validar valores frente a vocabularios específicos.

El Resource Description Framework anima a utilizar URI (*Uniform Resource Identifier*¹⁵) (identificador uniforme de recurso) como valores de propiedad, y muchos esquemas XML necesitan que los valores de atributo sean URI. Por ejemplo, en *XML-Signature Syntax and Processing* (sintaxis y proceso de la firma XML o *XMLDsig*), el valor del algoritmo del método de la firma debe ser un URI como «<http://www.w3.org/2000/09/xmldsig#dsa-sha1>».

En general, el URI es un valor permitido para las unidades semánticas del Diccionario de Datos PREMIS, salvo que exista alguna restricción. Sin embargo, el grupo de trabajo fue cauteloso al recomendar esta práctica. La gestión de URI

depende de un protocolo que, a pesar de su omnipresencia, está fuera del control del repositorio de preservación. Además, el grupo estaba convencido de que cualquier información requerida para la preservación a largo plazo debería almacenarse en el propio repositorio. Si esta información se almacena como un objeto de preservación, el *objectIdentifier* (identificador del objeto) del repositorio podrá hacer referencia a ella con mayor facilidad. La información almacenada de otra manera debería estar bajo el control directo del repositorio. Por lo tanto, la mayoría de los ejemplos del Diccionario de Datos son nombres de valores, en lugar de recursos URI. El equivalente al ejemplo anterior podría ser sencillamente «DSA-SHA1», que se define como una constante cuyo significado ya es conocido por el repositorio mediante alguna tabla u otro documento bajo el control de su organización.

1.4.5 Capacidad de extensión

El Diccionario de Datos indica el potencial de extensión de varias unidades semánticas, lo que permite incluir, en caso de ser necesario, más metadatos locales o proporcionar a los metadatos una estructura o un nivel de detalle mayor. La inclusión de estos metadatos adicionales es relativamente sencilla cuando se utilizan bases de datos relacionales; sin embargo, la primera versión del Diccionario de Datos y los esquemas PREMIS no ofrecían ningún mecanismo para añadir estos metadatos al utilizar dichos esquemas. La versión 2.0 del Diccionario de Datos introduce un mecanismo formal de extensión dentro de los esquemas para un reducido número de unidades semánticas, consideradas las más aptas para dicha extensión. En las siguientes actualizaciones del Diccionario de Datos se podría aumentar este conjunto inicial de unidades semánticas extensibles, si las circunstancias lo exigieran.

El conjunto inicial de unidades semánticas con capacidad de extensión son:

- *significantProperties* [Object entity]
- *objectCharacteristics* [Object entity]
- *creatingApplication* [dentro de *objectCharacteristics*, Object entity]
- *environment* [dentro de *objectCharacteristics*, Object entity]
- *signatureInformation* [Object entity]
- *eventOutcomeDetail* [dentro de *eventOutcomeInformation*, Event entity]

- rights [Rights entity]

Aplicando el contenedor de extensión que figura en el Diccionario de Datos estas unidades semánticas se pueden ampliar. En el Diccionario de Datos, dentro de los componentes semánticos definidos para cada una de las unidades semánticas enumeradas más arriba se indica la existencia de una unidad semántica que ejerce como contenedor de extensión y que llevará el término *extensión* añadido al nombre del contenedor que amplía. Una extensión puede contener metadatos codificados según un esquema externo.

Por otra parte, también se ha creado una nueva unidad semántica contenedora *objectCharacteristicsExtension* (extensión de las características del objeto) dentro de la entidad Objeto que permite incluir metadatos técnicos específicos dentro de PREMIS.

Al idear el mecanismo de extensión, el Comité Editorial de PREMIS estableció que sólo podrían ampliarse las unidades semánticas contenedoras. Esto permitiría tanto el uso de unidades semánticas definidas dentro de PREMIS como de un contenedor para las unidades semánticas definidas fuera de PREMIS, lo que requería ciertos cambios estructurales (por ejemplo, añadir un contenedor) para permitir la extensión de *eventOutcomeDetail* (detalles sobre las consecuencias del acontecimiento).

Al aplicar el mecanismo de extensión a las unidades semánticas extensibles de la lista, deben tenerse en cuenta los siguientes principios:

- Un contenedor de **extensión** puede utilizarse tanto para complementar como para reemplazar unidades semánticas de PREMIS dentro del contenedor padre (es decir, el contenedor que incluye el contenedor de extensión).

La única excepción es *objectCharacteristicsExtension* (extensión de las características del objeto), que puede complementar solamente a *objectCharacteristics* (características del objeto).

- Un contenedor de **extensión** puede utilizarse con las unidades semánticas de PREMIS ya existentes, complementando las unidades semánticas de PREMIS con metadatos adicionales.
- Un contenedor de **extensión** se puede utilizar sin unidades semánticas de PREMIS ya existentes, sustituyendo de forma eficaz las unidades semánticas

de PREMIS por otros metadatos aplicables (excepto para *objectCharacteristicsExtension* [extensión de las características del objeto]).

- Cuando existe una relación de uno a uno entre los contenidos de un contenedor de **extensión** y una unidad semántica de PREMIS existente, la práctica recomendada consiste en utilizar las unidades semánticas de PREMIS en vez de sus equivalentes en la extensión. Sin embargo, los implementadores, si las circunstancias lo permiten, pueden optar por utilizar solamente la extensión.
- Si no se utiliza ninguna unidad semántica debería suprimirse en vez de incluir un elemento vacío del esquema.
- Si la información en un contenedor de **extensión** necesita asociarse explícitamente a una unidad de PREMIS, el contenedor padre se repetirá con la subunidad apropiada. Si se necesitan extensiones procedentes de diferentes esquemas, el contenedor padre también se deberá repetir. En este caso, el contenedor padre repetido puede incluir el contenedor de extensión con o sin otras unidades semánticas de PREMIS existentes para dicho contenedor padre.
- Cuando se emplea un contenedor de **extensión**, se debe declarar el esquema externo que se utiliza dentro de dicho contenedor.

1.4.6 Los formatos de fecha y hora en PREMIS

Todas las unidades semánticas que especifican el uso de fecha o de fecha y hora sugieren el empleo de una forma estructurada para facilitar el proceso automático. Al no descender a los detalles de implementación, el Diccionario de Datos no especifica el uso de ningún formato en particular. En algunos casos, es necesaria la utilización de convenciones que permitan expresar aspectos de un periodo de tiempo como pueda ser una fecha abierta o dudosa.

La versión 2.0 del esquema XML de PREMIS especifica formatos de fecha y hora y establece dichas convenciones. Se recomienda su uso cuando sea necesario. Las siguientes son unidades semánticas que pueden incluir la fecha o bien la fecha y la hora:

- `preservationLevelDateAssigned` (fecha asignada al nivel de preservación) (bajo `preservationLevel` [nivel de preservación])

- `dateCreatedByApplication` (fecha creada por la aplicación) (bajo `creatingApplication` [aplicación creadora])
- `eventDateTime` (fecha y hora del acontecimiento) (bajo `event` [acontecimiento])
- `copyrightStatusDeterminationDate` (determinación de la fecha del estado del copyright) (bajo `copyrightInformation` [información del copyright])
- `statuteInformationDeterminationDate` (fecha de determinación de la información sobre la legislación) (bajo `statuteInformation` [información sobre la legislación])
- `startDate` (fecha de inicio) (bajo `termOfGrant` [periodo por el que se otorgan los derechos])
- `endDate` (fecha final) (bajo `termOfGrant` [periodo por el que se otorgan los derechos])

2. EL DICCIONARIO DE DATOS PREMIS VERSIÓN 2.0

El Diccionario de Datos PREMIS incluye unidades semánticas para Objetos, Acontecimientos, Agentes y Derechos. En el modelo se muestra una quinta entidad, la "Entidad Intelectual" que no se incluye en el Diccionario ya que está bien analizada por los metadatos descriptivos. La estructura de cada entrada incluye notas sobre cómo crear o utilizar la unidad semántica. En algunos casos se consideró que sería útil disponer de información adicional para explicar los motivos por los que se redactó una definición de determinada manera o sobre temas que surgieron en las reuniones. Véase [Temas especiales](#).

Un componente semántico siempre hereda la aplicabilidad de la unidad semántica en la que está incluido. Es decir, que si la unidad semántica contenedora especifica que se puede aplicar a ficheros pero no a representaciones, cada uno de los componentes semánticos se aplica a ficheros y no a representaciones. Sin embargo, la obligatoriedad y el que sean repetibles o no, pueden variar.

Cada entrada del Diccionario de Datos presenta los siguientes atributos para cada unidad semántica:

- **Nombre de la unidad semántica:** los nombres se concibieron para que fueran descriptivos y únicos en el Diccionario de Datos. El uso de estos nombres favorece el intercambio de metadatos y la interoperabilidad entre repositorios. No es necesario utilizar estos mismos nombres en un repositorio de preservación.
- **Componentes semánticos:** cada componente semántico tiene su propia entrada en el Diccionario de Datos. Una unidad semántica con componentes no tiene asignados valores propios. Únicamente las unidades semánticas del nivel más bajo tienen valores asignados.
- **Definición:** el significado de la unidad semántica.
- **Justificación:** motivo por el que se necesita la unidad semántica en caso de que no sea evidente en la definición.

- **Limitaciones:** como debe codificarse el valor de la unidad semántica. Algunas de las limitaciones más comunes son:

Contenedor –La unidad semántica es un paraguas para dos o más componentes semánticos y no tiene valores propios.

Ninguno –La unidad semántica puede adoptar cualquier forma para el valor.

El valor debería tomarse de un vocabulario controlado –El repositorio de preservación debería establecer una lista de autoridades de aquellos valores que sean útiles y significativos para el repositorio. El Diccionario de Datos PREMIS no especifica qué listado de autoridades hay que utilizar, asume que cada repositorio utilizará su propio vocabulario. Como norma general, cuando el valor se tome de un vocabulario controlado debería consignarse la fuente. Puede registrarse la fuente utilizando los esquemas XML para PREMIS.

- **Categoría del objeto:** indica si la unidad es válida para una representación, para un fichero o para una cadena de bits.

Las unidades semánticas que son válidas para los ficheros también lo son para las [cadenas de bits](#).

- **Aplicabilidad:** indica si el valor es aplicable a una determinada categoría de Objeto

- **Ejemplos:** Uno o varios ejemplos de los valores que pueden aparecer en la unidad semántica. Se intenta que sean ilustrativos.

En texto plano se muestra el nombre de un valor real y en texto entre corchetes está la descripción de ese valor. Por ejemplo, "SHA-1 message digest" es el valor para la unidad semántica, y "[SHA-1 message digest]" quiere decir que el valor que representa la unidad semántica es un mensaje cifrado (message digest) SHA-1 como "7c9b35da4f2ebd436f1cf88e5a39b3a257edf4a22be3c955ac49da2e2107b67a1924419563"

- **Repetibilidad:** Una unidad semántica designada como "Repetible" puede aparecer varias veces mostrando valores diferentes. Esto no significa que un repositorio deba registrar varias unidades semánticas iguales.

- **Obligatoriedad:** una unidad semántica obligatoria contiene información que el repositorio de preservación necesita conocer con independencia de

cuándo y cómo se registra dicha información. No es necesario registrar explícitamente el valor de la unidad semántica si éste se puede obtener por otros medios (por ejemplo, por las reglas de negocio –business rules- del repositorio). “Obligatorio” en realidad significa “Obligatorio si corresponde”. Por ejemplo, el identificador de una cadena de bits es obligatorio (M)¹ sólo si el repositorio gestiona datos a nivel de cadena de bits. A la hora de intercambiar metadatos PREMIS entre repositorios los valores que corresponden a unidades semánticas obligatorias siempre han de proporcionarse.

Se recomienda el uso de unidades semánticas “Opcionales” (O) pero no son necesarios.

Si el contenedor es opcional (O) pero uno de sus componentes semánticos es obligatorio (M), dicho componente se debe proporcionar única y exclusivamente si el contenedor se registra. Es decir si se proporciona un valor, ya sea opcional u obligatorio para la unidad semántica contenedora entonces debe registrarse el valor de todas las unidades obligatorias incluidas en el contenedor.

- **Notas de creación y/o mantenimiento:** notas acerca de la obtención y la actualización de los valores de las unidades semánticas.
- **Notas de uso:** información relativa al posible uso de la unidad semántica o una aclaración sobre la definición.

2.1 Alcance del Diccionario de Datos

Metadatos descriptivos:

Normalmente, los metadatos descriptivos se utilizan para describir Entidades Intelectuales.

Casi todos los repositorios de preservación incluyen metadatos descriptivos o enlaces a los metadatos descriptivos localizados fuera del propio repositorio. Estos metadatos pueden identificar un recurso mediante los datos de publicación como el autor y el título, o pueden identificar su contenido intelectual mediante clasificaciones temáticas, materias, etc. Los metadatos descriptivos pueden ser importantes tanto para localizar los recursos archivados como para ayudar a tomar decisiones durante la planificación de la preservación.

¹ En la traducción se ha decidido respetar la sigla utilizada en la versión original: Data Dictionary for Preservation Metadata: PREMIS version 2.0

Sin embargo, el Diccionario de Datos no se centra en los elementos descriptivos por dos motivos. El primero es que los metadatos descriptivos ya han sido definidos por otros estándares existentes como los esquemas MARC¹⁶, MODS¹⁷, el Dublin Core Metadata Element Set¹¹, el Content Standard for Digital Geospatial Metadata¹⁸, el VRA Core¹⁹, el Encoded Archival Description (EAD)²⁰ y el Data Documentation Initiative²¹ los cuales son sólo algunos ejemplos de estándares que definen los elementos de metadatos descriptivos. El grupo de trabajo no quiso añadir otro conjunto de elementos descriptivos a un campo ya cubierto.

El segundo motivo es que, a menudo, los metadatos descriptivos dependen de cada dominio. Para efectos de preservación es menos importante que un conjunto de elementos describa, por ejemplo, telemetría satelital u obras de Picasso digitalizadas, que el hecho de que grupos con intereses comunes puedan capturar e intercambiar información de forma que queden reflejados correctamente sus propios materiales e intereses.

Agentes: PREMIS no define con detalle las características de los Agentes. Los metadatos que describen personas, organizaciones y otras entidades que pueden actuar como Agentes han sido definidos en muchos formatos y estándares existentes como, por ejemplo, MARC¹⁶, vCard²², MADS²³ y otros esquemas en desarrollo. Siempre que un repositorio de preservación pueda identificar correctamente los Agentes que han actuado sobre los Objetos que almacena, las características adicionales de los Agentes vendrán determinadas por los requisitos concretos en cada caso; muchas de las características adicionales de los Agentes pueden definirse a partir de conjuntos estándar de elementos de metadatos.

Derechos: PREMIS se centra en definir principalmente las características de los derechos y permisos relacionados con actividades de preservación, y omite aquellos relacionados con el acceso o la difusión. Esta revisión del Diccionario amplía las unidades semánticas utilizadas para la información de derechos y permite agregar un esquema externo de metadatos de derechos.

Metadatos técnicos: Los metadatos técnicos describen más las características físicas que las características intelectuales de los objetos digitales. Los metadatos técnicos detallados y con formato específico son necesarios para poner en práctica la mayoría de estrategias de preservación, pero el grupo no disponía ni del tiempo ni de los conocimientos suficientes para profundizar en los metadatos técnicos específicos de cada formato o tipo de fichero digital. Por lo tanto, redujo los metadatos técnicos incluidos en Diccionario de Datos a las unidades semánticas que

creían que podían aplicarse a todo tipo de objetos fuera cual fuese su formato. Se prefiere que sean los expertos en formatos quienes proporcionen un desarrollo más profundo de los metadatos técnicos. Se proporciona un mecanismo de extensión al incluir la unidad semántica *objectCharacteristicsExtension* (extensión de las características del objeto), que se puede utilizar con un esquema externo de metadatos técnicos.

Detalles sobre hardware o soportes: El grupo de trabajo no trató de definir los metadatos para documentar con detalle los soportes y el hardware. Por ejemplo, PREMIS define una unidad semántica para identificar el soporte en el que se almacena un objeto. Un repositorio de preservación probablemente estará interesado en conocer más detalles acerca de los soportes empleados. Si el repositorio almacena datos en DVD, por ejemplo, puede que necesite conocer con detalle las características técnicas de las unidades de DVD específicas, como el fabricante, el material utilizado para colorear el DVD y su densidad. PREMIS reserva a los especialistas en estas áreas la tarea de definir los metadatos que describen las características de los soportes y el hardware.

Reglas de negocios: El grupo de trabajo no trató de describir las reglas de negocio de los repositorios, aunque estos metadatos sean esenciales para la preservación dentro de éstos.

Las reglas de negocio codifican las estrategias de preservación de la aplicación y documentan las políticas, los servicios, los cargos y las funciones del repositorio. Cuestiones como los períodos de retención, la disposición, la valoración de riesgos, los promedios de permanencia, la programación de actualización de los soportes, entre otras, guardan relación con los Objetos pero no constituyen propiedades reales de éstos. Se hizo una excepción para el nivel del tratamiento de preservación de un objeto (*preservationLevel* [nivel de preservación]) por su información crítica para cualquier repositorio de preservación. Se podría haber añadido un tratamiento más riguroso de las reglas de negocio al modelo de datos definiendo una entidad de reglas similar a la de los Derechos, pero no se ha hecho para la versión actual.

2.2 Entidad Objeto

La entidad Objeto contiene información acerca de un objeto digital custodiado en un repositorio y describe las características del objeto relevantes para la preservación.

ObjectIdentifier es la única unidad semántica obligatoria que afecta a todas las categorías de objetos (representación, fichero y cadena de bits).

2.2.1 Tipos de Entidad

- **Representación:** objeto digital que materializa o representa una Entidad intelectual. Una representación es el conjunto de Ficheros y metadatos estructurales almacenados y necesarios para conseguir una reproducción completa y con sentido de una Entidad intelectual.
- **Fichero:** secuencia de bytes nombrada y ordenada que puede ser reconocida por un sistema operativo
- **Cadena de bits:** Datos de un fichero, contiguos o no contiguos, con propiedades comunes a efectos de preservación.

2.2.2 Propiedades de la Entidad

- Puede asociarse a una o a más declaraciones de derechos.
- Puede participar en uno o en más acontecimientos.
- Los enlaces entre entidades pueden registrarse desde cualquiera de los sentidos y no tienen por qué ser bidireccionales.

2.2.3 Unidades semánticas de la Entidad

1.1 *objectIdentifier* (identificador del objeto) (O, R)

1.1.1 *objectIdentifierType* (tipo del identificador del objeto) (O, NR)

1.1.2 *objectIdentifierValue* (Valor del identificador del objeto) (O, NR)

1.2 *objectCategory* (categoría del objeto) (O, NR)

1.3 *preservationLevel* (nivel de preservación) (NO, R) [representación, fichero]

1.3.1 *preservationLevelValue* (valor del nivel de preservación) (O, NR)
[representación, fichero]

- 1.3.2 preservationLevelRole (función del nivel de preservación) (M, NR) [representación, fichero]
- 1.3.3 preservationLevelRationale (fundamentos del nivel de preservación) (M, R) [representación, fichero]
- 1.3.4 preservationLevelDateAssigned (fecha asignada al nivel de preservación) (M, NR) [representación, fichero]
- 1.4 significantProperties (propiedades significativas) (M, R)
 - 1.4.1 significantPropertiesType (tipo de propiedades significativas) (M, NR)
 - 1.4.2 significantPropertiesValue (valor de las propiedades significativas) (M, NR)
 - 1.4.3 significantPropertiesExtension (extensión de las propiedades significativas) (M, R)
- 1.5 objectCharacteristics (características del objeto) (O, R) (fichero, cadena de bits)
 - 1.5.1 compositionLevel (nivel de composición) (O, NR) (fichero, cadena de bits)
 - 1.5.2 fixity (fijeza) (M, R) (fichero, cadena de bits)
 - 1.5.2.1 messageDigestAlgorithm (algoritmo del mensaje cifrado) (O, NR) (fichero, cadena de bits)
 - 1.5.2.2 messageDigest (mensaje cifrado) (O, NR) [fichero, cadena de bits]
 - 1.5.2.3 messageDigestOriginator (creador del mensaje cifrado) (M, NR) [fichero, cadena de bits]
 - 1.5.3 size (tamaño)(O, NR) [file, bitstream]
 - 1.5.4 format (formato) (O, R) [fichero, cadena de bits]
 - 1.5.4.1 formatDesignation (designación del formato) (M, NR) [fichero, cadena de bits]
 - 1.5.4.1.1 formatName (nombre del formato) (O, NR) [fichero, cadena de bits]
 - 1.5.4.1.2 formatVersion (versión del formato) (M, NR) [fichero, cadena de bits]

1.5.4.2 formatRegistry (registro del formato) (M, NR) [fichero, cadena de bits]

1.5.4.2.1 formatRegistryName (nombre del registro del formato) (O, NR) [fichero, cadena de bits]

1.5.4.2.2 formatRegistryKey (clave del registro del formato) (O, NR) [fichero, cadena de bits]

1.5.4.2.3 formatRegistryRole (función del registro del formato) (M, NR) [fichero, cadena de bits]

1.5.4.3 formatNote (nota sobre el formato) (M, R) [fichero, cadena de bits]

1.5.5 creatingApplication (aplicación creadora) (M, R) [fichero, cadena de bits]

1.5.5.1 creatingApplicationName (nombre de la aplicación creadora) (M, NR) [fichero, cadena de bits]

1.5.5.2 creatingApplicationVersion (versión de la aplicación creadora) (M, NR) [fichero, cadena de bits]

1.5.5.3 dateCreatedByApplication (fecha creada por la aplicación) (M, NR) [fichero, cadena de bits]

1.5.5.4 creatingApplicationExtension (extensión de la aplicación creadora) (M, R) [fichero, cadena de bits]

1.5.6 inhibitors (inhibidores) (M, R) [fichero, cadena de bits]

1.5.6.1 inhibitorType (tipo de inhibidor) (O, NR) [fichero, cadena de bits]

1.5.6.2 inhibitorTarget (objetivo del inhibidor) (M, R) [fichero, cadena de bits]

1.5.6.3 inhibitorKey (clave del inhibidor) (M, NR) [fichero, cadena de bits]

1.5.7 objectCharacteristicsExtension (extensión de las características del objeto) (M, R) [fichero, cadena de bits]

1.6 originalName (nombre original) (M, NR) [representación, fichero]

1.7 storage (almacenamiento) (O, R) [fichero, cadena de bits]

1.7.1 contentLocation (localización del contenido) (M, NR) [fichero, cadena de bits]

1.7.1.1 contentLocationType (tipo de localización del contenido) (O, NR) [fichero, cadena de bits]

1.7.1.2 contentLocationValue (valor de la localización del contenido) (O, NR) [fichero, cadena de bits]

1.7.2 storageMedium (soporte del almacenamiento) (M, NR) [fichero, cadena de bits]

1.8 environment (entorno) (M, R)

1.8.1 environmentCharacteristics (características del entorno) (M, NR)

1.8.2 environmentPurpose (propósito del entorno) (M, R)

1.8.3 environmentNote (nota sobre el entorno) (M, R)

1.8.4 dependency (dependencia) (M, R)

1.8.4.1 dependencyName (nombre de la dependencia) (M, R)

1.8.4.2 dependencyIdentifier (identificador de la dependencia) (M, R)

1.8.4.2.1 dependencyIdentifierType (tipo de identificador de dependencia) (O, NR)

1.8.4.2.2 dependencyIdentifierValue (valor del identificador de dependencia) (O, NR)

1.8.5 software (M, R)

1.8.5.1 swName (nombre del software) (O, NR)

1.8.5.2 swVersion (versión del software) (M, NR)

1.8.5.3 swType (tipo de software) (O, NR)

1.8.5.4 swOtherInformation (otra información sobre el software) (M, R)

1.8.5.5 swDependency (dependencia del software) (M, R)

1.8.6 hardware (M, R)

1.8.6.1 hwName (nombre del hardware) (O, NR)

1.8.6.2 hwType (tipo de hardware) (O, NR)

1.8.6.3 hwOtherInformation (otra información sobre el hardware) (M, R)

1.8.7 environmentExtension (extensión del entorno) (M, R)

1.9 signatureInformation (información sobre la firma) (M, R) [fichero, cadena de bits]

1.9.1 signature (firma) (M, R)

1.9.1.1 signatureEncoding (codificación de la firma) (O, NR) [fichero, cadena de bits]

1.9.1.2 signer (firmante) (M, NR) [fichero, cadena de bits]

1.9.1.3 signatureMethod (método de la firma) (O, NR) [fichero, cadena de bits]

1.9.1.4 signatureValue (valor de la firma) (O, NR) [fichero, cadena de bits]

1.9.1.5 signatureValidationRules (reglas de validación de la firma) (O, NR) [fichero, cadena de bits]

1.9.1.6 signatureProperties (propiedades de la firma) (M, R) [fichero, cadena de bits]

1.9.1.7 keyInformation (información sobre la clave) (M, NR) [fichero, cadena de bits]

1.9.2 signatureInformationExtension (extensión de la información sobre la signatura) (M, R) [fichero, cadena de bits]

1.10 relationship (relaciones) (M, R)

1.10.1 relationshipType (tipo de relaciones) (O, NR)

1.10.2 relationshipSubType (subtipo de relaciones) (O, NR)

1.10.3 relatedObjectIdentification (identificación del objeto relacionado) (O, R)

1.10.3.1 relatedObjectIdentifierType (tipo de identificador del objeto relacionado) (O, NR)

1.10.3.2 relatedObjectIdentifierValue (valor del identificador del objeto relacionado) (O, NR)

1.10.3.3 relatedObjectSequence (secuencia del objeto relacionado)
(M, NR)

1.10.4 relatedEventIdentification (identificación del acontecimiento
relacionado) (M, R)

1.10.4.1 relatedEventIdentifierType (tipo de identificador del
acontecimiento relacionado) (O, NR)

1.10.4.2 relatedEventIdentifierValue (valor del identificador del
acontecimiento relacionado) (O, NR)

1.10.4.3 relatedEventSequence (secuencia del acontecimiento
relacionado) (M, NR)

1.11 linkingEventIdentifier (identificador del acontecimiento vinculado) (M, R)

1.11.1 linkingEventIdentifierType (tipo de identificador del acontecimiento
vinculado) (O, NR)

1.11.2 linkingEventIdentifierValue (valor del identificador del acontecimiento
vinculado) (O, NR)

1.12 linkingIntellectualEntityIdentifier (identificador de la entidad intelectual
vinculada) (M, R)

1.12.1 linkingIntellectualEntityIdentifierType (tipo de identificador de la
entidad intelectual vinculada) (O, NR)

1.12.2 linkingIntellectualEntityIdentifierValue (valor del identificador de la
entidad intelectual vinculada) (O, NR)

1.13 linkingRightsStatementIdentifier (identificador de la mención de derechos
vinculada) (M, R)

1.13.1 linkingRightsStatementIdentifierType (tipo de identificador de la
mención de derechos vinculada) (O, NR)

1.13.2 linkingRightsStatementIdentifierValue (valor del identificador de la
mención de derechos vinculada) (O, NR)

Unidad semántica	1.1 objectIdentifier		
Componentes semánticos	1.1.1 objectIdentifierType 1.1.2 objectIdentifierValue		
Definición	Elemento de identificación unívoca del objeto digital en el sistema de preservación		
Justificación	Texto (Estilo. Texto de tabla)		
Limitaciones	Texto (Estilo. Texto de tabla)		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Notas de creación / mantenimiento	<p>El identificador puede ser creado por el sistema en el momento del proceso de carga de objetos digitales (ingest) o puede ser creado o asignado de forma externa. y ser incluido como metadato junto con el objeto. De la misma manera los identificadores pueden ser generados de forma automática o de forma manual. Se recomienda que los repositorios utilicen identificadores creados automáticamente por el repositorio como identificadores principales para asegurar el que el identificador sea único. Los identificadores asignados de forma externa pueden utilizarse para asociar el objeto a la información externa.</p>		
Notas de uso	<p><i>objectIdentifier</i> es obligatorio si el repositorio almacena y gestiona objetos a ese nivel (i.e., representación, fichero, cadena de bits) <i>objectIdentifier</i> es repetible para permitir los dos tipos de identificadores. Véase la nota de creación o mantenimiento Los identificadores deben ser únicos en el repositorio. Pueden haber sido creados con anterioridad y ser utilizados en otros sistemas de gestión de objetos digitales. Los identificadores que se utilizan para identificar una clase de objetos (por ejemplo como el ISBN identifica todos los libros de la misma edición) no son recomendables para un repositorio ya que este debe identificar el objeto específico que alberga. Un repositorio necesita conocer tanto el tipo de identificador del objeto como el valor. Si el valor contiene el tipo de identificador (ej.: "oai:lib.uchicago.edu:1"), el identificador puede omitirse. Igualmente si el repositorio utiliza solo un tipo de identificador puede omitirse. Es recomendable utilizar un identificador permanente pero el esquema concreto de identificadores es una decisión específica de la implementación.</p>		

Unidad semántica	1.1.1 objectIdentifierType		
Componentes semánticos	Ninguno		
Definición	Denominación del dominio en el que el objeto es único		
Justificación	No podemos asumir que los valores del Identificador sean únicos fuera del repositorio. La combinación entre objectIdentifierType y objectIdentifierValue debería garantizar su unicidad.		
Limitaciones	El valor debe proceder de un listado controlado de términos		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	DLC DRS hdl:4263537	DLC DRS hdl:4263537	DLC DRS hdl:4263537
Notas de uso	El tipo del identificador debe formar parte implícita del repositorio siempre que pueda transmitirse cuando el objeto se exporte		

Unidad semántica	1.1.2 objectIdentifierValue		
Componentes semánticos	Ninguno		
Definición	Valor del Identificador de objectIdentifier.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No repetible	No repetible	No repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	00000000312	IU2440 http://nrs.harvard.edu/urn-3:FHCL.Loeb:sa1	IU2440-1 IU2440-2

Unidad semántica	1.2 objectCategory		
Componentes semánticos	Ninguno		
Definición	Categoría del objeto al cual se asocian los metadatos		
Justificación	Los repositorios deberían tratar las diferentes categorías de objetos (representaciones, ficheros y cadenas de bits) de forma diferente para los metadatos y para las funciones de gestión		
Limitaciones	El valor debe proceder de un listado controlado de términos		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No repetible	No repetible	No repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	Representación	Archivo	Cadena de bits
Notas de uso	Valores propuestos: representación, fichero, cadena de bits. Un filestream podría tratarse como un fichero		

Unidad semántica	1.3 preservationLevel		
Componentes semánticos	1.3.1 preservationLevelValue 1.3.2 preservationLevelRole 1.3.3 preservationLevelRationale 1.3.4 preservationLevelDataAssigned		
Definición	Información referente a la política o nivel de preservación aplicable a cada objeto y el contexto a que se refiere.		
Justificación	Algunos repositorios de preservación ofrecerán múltiples opciones de preservación dependiendo de factores como el valor o la singularidad del material, la capacidad del formato para ser preservado, la suma que el usuario está dispuesto a pagar, etc. El contexto que rodea a la elección de una opción determinada también puede requerir explicaciones adicionales.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	No aplicable
Repetible	Repetible	Repetible	
Obligatoriedad	Opcional	Opcional	
Notas de uso	Si el repositorio sólo ofrece un nivel de preservación no es necesario registrar este valor. La aplicación de un conjunto determinado de unidades semánticas <i>preservationLevel</i> puede abarcar sólo una representación de un objeto. Las representaciones en otras formas técnicas o para otras funciones pueden recibir un <i>preservationLevel</i> diferente. El contenedor puede repetirse si se necesita registrar un valor <i>preservation Level</i> en otros contextos (Véase preservationLevelRole).		

Unidad semántica	1.3.1 preservationLevelValue		
Componentes semánticos	Ninguno		
Definición	Un valor que indica el conjunto de funciones de preservación que se deberían aplicar al objeto.		
Justificación	Algunos repositorios de preservación ofrecerán múltiples opciones de preservación dependiendo de factores como el valor o la singularidad del material, la capacidad del formato para ser preservado, la suma que el usuario está dispuesto a pagar, etc.		
Limitaciones	El valor debe proceder de un listado controlado de términos.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	No Aplicable
Repetible	No Repetible	No repetible	
Obligatoriedad	Obligatorio	Obligatorio	
Ejemplo	Bit level Full 0 1 2	Bit level Full 0 Plenamente compatible con futuras migraciones	
Notas de uso	Sólo debe registrarse un <i>preservationLevelValue</i> por cada contenedor <i>PreservationLevel</i> . Si en un contexto distinto puede aplicarse otro <i>preservationLevelValue</i> debe repetirse un contenedor <i>preservationLevel</i> separado.		

Unidad semántica	1.3.2 preservationLevelRole		
Componentes semánticos	Ninguno		
Definición	Un valor que indica el contexto en el que es posible aplicar un conjunto de estrategias de preservación.		
Justificación	Como hemos visto pueden existir diferentes <i>preservationLevelValues</i> en un repositorio y en ocasiones es necesario registrar más de un valor según cada contexto.		
Limitaciones	El valor debe proceder de un listado controlado de términos.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	No aplicable
Repetible	No repetible	No repetible	
Obligatoriedad	Opcional	Opcional	
Ejemplo	Requirement = exigencia Intention = intención Capability = capacidad	Requirement = exigencia Intention = intención Capability = capacidad	
Notas de uso	<p>Esta unidad semántica opcional especifica el motivo o el contexto en el que <i>preservationLevelValue</i> se aplica al actual <i>preservationLevel</i>.</p> <p>Por ejemplo un repositorio puede tener la obligación legal de preservar de forma completa un objeto X (que tiene un formato F) pero actualmente sólo puede preservar objetos de formato F a nivel de bits. El repositorio puede necesitar registrar tanto el nivel de preservación exigido (e.g. <i>preservationLevelRole</i>="requirement") como la capacidad actual. (e.g. <i>preservationLevelRole</i>="capability").</p> <p>En el momento de pasar la custodia de un material de un repositorio a otro puede ser importante para el receptor conocer el sentido en el que <i>preservationLevelValue</i> debería entenderse. El repositorio receptor no necesita conocer la capacidad potencial de preservación del repositorio de procedencia (ya que esto no influye en su capacidad) pero si debe conocer cualquier nivel de preservación exigido para el material del que va a responsabilizarse</p> <p>Se considera buena práctica el especificar <i>preservationLevelRole</i> como aclaración aunque el repositorio sólo asigne <i>preservationLevelValue</i> en un contexto. Si se consigna más de un <i>preservationLevel</i>, siempre debe proporcionarse <i>preservationLevelRole</i>.</p> <p>Si es necesario indicar más de un contexto para el mismo objeto (ej. se consignan tanto "exigencia" como "capacidad") deben utilizarse contenedores <i>preservationLevel</i> separados.</p>		

Unidad semántica	1.3.3 preservationLevelRationale		
Componentes semánticos	Ninguno		
Definición	Motivo por el que se asigna al objeto un <i>preservationLevelValue</i> determinado.		
Justificación	Aplicar un <i>preservationLevelValue</i> determinado puede exigir justificación, sobre todo si éste es distinto del que normalmente se aplica en el repositorio.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	No aplicable
Repetible	Repetible	Repetible	
Obligatoriedad	Opcional	Opcional	
Ejemplo	Presupuesto legislación	Archivo defectuoso Sólo es posible aplicar un bajo nivel de preservación a este formato	
Notas de uso	<p>Esta unidad semántica opcional registra el motivo de la aplicación de <i>preservationLevelValue</i>.</p> <p>Esta información puede ser particularmente importante cuando <i>preservationLevelValue</i> es diferente del utilizado normalmente.</p> <p>Por ejemplo, un repositorio puede asignar normalmente un valor <i>preservationLevelValue</i> de "full preservation" para ficheros JPEG2000 pero detecta que un fichero en concreto es defectuoso. Esto significaría que la estrategia de preservación del repositorio para JPEG2000 no sería efectiva en el caso de este fichero, por lo tanto el repositorio puede asignar a este fichero un <i>preservationLevelValue</i> de "preservación a nivel de bit" registrando como motivo "fichero defectuoso".</p> <p>También hay requerimientos legales o convenios que exigen aplicar un nivel de preservación mayor del habitual a un objeto en concreto. En estos casos el motivo puede registrarse por ejemplo como: "legislación" o "el usuario paga"</p> <p><i>preservationLevelRationale</i> puede repetirse si hay que consignar más de un motivo.</p>		

Unidad semántica	1.3.4 preservationLevelDataAssigned		
Componentes semánticos	Ninguno		
Definición	La fecha o la fecha y la hora en las que se asignó a un objeto un <code>preservationLevelValue</code> determinado.		
Justificación	Se considera que el <code>preservationLevel</code> aplicado a un objeto digital debe ser revisado y en consecuencia modificado con el tiempo en consonancia a la evolución tecnológica, los cambios políticos, y el interés del objeto digital. El conocer la fecha de asignación del <code>preservationLevel</code> ayudará en la toma de decisiones posteriores		
Limitaciones	Este valor debería adoptar una forma estructurada para facilitar el proceso automatizado de los datos. Se recomienda utilizar convenciones normalizadas que faciliten el intercambio de metadatos conforme a PREMIS, por ejemplo las utilizadas en el esquema PREMIS.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	No aplicable
Repetible	No Repetible	No Repetible	
Obligatoriedad	Opcional	Opcional	
Ejemplo	2007-11-05 2007-11-05 T08:15:30-05:00 20080315	2007-11-05 2007-11-05 T08:15:30-05:00 20080315	

Unidad semántica	1.4 significantProperties		
Componentes semánticos	1.4.1 significantPropertiesType 1.4.2 significantPropertiesValue 1.4.3 significantPropertiesExtension		
Definición	Características de un objeto concreto que se determina subjetivamente que se mantengan durante todo el proceso		
Justificación	Objetos que tienen las mismas propiedades técnicas pueden diferir en cuanto a requisitos de preservación necesarios para asegurar futuras presentaciones o usos del objeto digital.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas de creación /mantenimiento	<p>Los objetos digitales en un mismo formato comparten propiedades, por ejemplo, el responsable del repositorio puede decidir que para todos los ficheros PDF tan sólo sea necesaria la preservación del contenido. Mientras que para los objetos artísticos será necesario definir las propiedades exclusivas de cada documento.</p> <p>Cuando los valores son únicos deben ser proporcionados por quien entrega el objeto o proporcionados por el personal técnico a cargo del repositorio.</p>		
Notas de uso	<p>Todas estas subunidades de la unidad semántica son opcionales. Si se utiliza este contenedor debe estar presente por lo menos una de las subunidades <i>significantPropertiesValue</i> o <i>significantPropertiesExtension</i>.</p> <p>Las propiedades significativas pueden ser unas características técnicas que se consideren importantes o unas características determinadas subjetivamente. Por ejemplo un PDF puede incluir enlaces que no se consideren importantes y JavaScript que si se considere importante. Las migraciones futuras de una imagen TIFF pueden requerir un tratamiento de optimización para la claridad de la línea o para el color; la opción que se elija dependerá del juicio crítico del conservador sobre las propiedades significativas de la imagen.</p> <p>Consignar las propiedades significativas implica que el repositorio planea preservar estas propiedades y pretende que superen de forma aceptable las acciones de preservación; por ejemplo, durante la emulación o después de una migración de formato. También implica que el repositorio anotará si las acciones de preservación provocan una modificación de las propiedades.</p> <p>En la práctica las propiedades significativas podrían usarse como indicadores del éxito de la preservación, como parte del control de calidad de los resultados de una acción de preservación o de la evaluación de la eficacia de un método de preservación. Por ejemplo, si la lista de propiedades no se mantiene tras la aplicación de un método determinado de preservación podría</p>		

	<p>indicar un fallo del proceso o que el método no es el adecuado para ese material.</p> <p>Se necesita más experiencia con la preservación digital para determinar la mejor manera de representar las propiedades esenciales y sus modificaciones.</p> <p>Las unidades semánticas incluidas en el contenedor <i>significantProperties</i> pretenden proporcionar una estructura flexible para describir propiedades significativas, permitiendo que tipos genéricos de aspectos, facetas o atributos de un objeto se pongan de manifiesto y se emparejen con detalles específicos y significativos sobre el objeto en relación con ese aspecto, faceta o atributo.</p> <p>Por ejemplo, algunos repositorios pueden definir propiedades significativas para objetos relacionadas con facetas de contenido, apariencia, estructura, comportamiento y contexto.</p> <p>Ejemplos de facetas: en este caso las parejas pueden incluir en detalle:</p> <p><i>significantPropertiesType</i> = "content" "contenido" <i>significantPropertiesValue</i> = "all textual content and images" "todo el contenido textual e imágenes"</p> <p><i>significantPropertiesType</i> = "behavior" "comportamiento" <i>significantPropertiesValue</i> = "editable" "editable"</p> <p>Otros repositorios pueden optar por describir las propiedades significativas a un nivel de atributos más granular: por ejemplo</p> <p><i>significantPropertiesType</i> = "page count" "número de páginas" <i>significantPropertiesValue</i> = "7"</p> <p><i>significantPropertiesType</i> = "page width" "ancho de página" <i>significantPropertiesValue</i> = "210 mm"</p> <p>Cada faceta: la pareja de detalles debería incluirse en un contenedor <i>significantProperties</i> separado y repetido.</p> <p>Los trabajos que se realicen para determinar y describir propiedades significativas producirán esquemas más detallados que facilitarán las descripciones.</p> <p>También requiere estudios más detallados la representación de las modificaciones de las propiedades significativas a consecuencia de una acción de preservación. Una de las posibilidades implica el uso de información de Objeto y de Evento: El Objeto A tiene como propiedades significativas volumen y sincronización que se registran como <i>significantProperties</i> de A. En la versión migrada B, la sincronización se modifica, lo que se anota en <i>eventOutcome</i> del evento migración. Sólo el volumen se anota como propiedad significativa de B.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Unidad semántica	1.4.1 significantPropertiesType		
Componentes semánticos	Ninguno		
Definición	Aspecto, clase o atributo de un objeto del que se describen propiedades significativas.		
Justificación	Los repositorios pueden optar por describir propiedades significativas basadas en un aspecto particular o un atributo de un objeto.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo	contenido estructura comportamiento número de página ancho de página tipo de letra enlaces número de imagen	contenido estructura comportamiento número de página ancho de página tipo de letra	[para una imagen embebida] espacio de color
Notas de uso	Esta unidad semántica es opcional y puede utilizarse como parte de una faceta: par detallada junto con <i>significantPropertiesValue</i> .		

Unidad semántica	1.4.2 significantPropertiesValue		
Componentes semánticos	Ninguno		
Definición	Descripción de las características de un objeto que potencialmente puedan ser importantes para la preservación del objeto.		
Justificación	El responsable del repositorio puede optar por describir propiedades importantes de un objeto específicas o exclusivas del mismo.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo	Sólo contenido [para una página Web que contenga animaciones que no se consideren esenciales]. "hiperenlaces " [Para detalle asociado con un <i>significantProperties Type</i> "comportamiento"]	Sólo contenido [para un documento en word con enlaces que no se consideren esenciales] Editable [Para detalle asociado con un <i>significantProperties Type</i> "comportamiento"] 210mm [Para detalle asociado con <i>significantProperties Type</i> "ancho de página"] 210 mm	Color [para un PDF que incluya un gráfico en el que el color de las líneas determine el significado] Color. [Para detalle asociado con <i>significantProperties Type</i> de"apariencia"]
Notas sobre uso	Si se utilizan los pares faceta:detalle el contenido de <i>significantPropertiesValue</i> debería describir las propiedades significativas del objeto relevantes para el aspecto, la faceta o los atributos que se han declarado en <i>significantPropertiesType</i> con el que está emparejado. Si no se utilizan los pares faceta:detalle <i>significantPropertiesValue</i> puede utilizarse para describir libremente cualquier característica de un objeto. <i>significantPropertiesValue</i> no es repetible. Si hay muchas propiedades significativas deberían describirse en unidades contenedoras <i>significantProperties</i> separadas y repetidas.		

Unidad semántica	1.4.3 significantPropertiesExtension		
Componentes semánticos	Definidos externamente		
Definición	Contenedor que incluye unidades semánticas definidas fuera del Diccionario de Datos PREMIS.		
Justificación	En ocasiones puede ser necesario reemplazar o ampliar unidades semánticas definidas en los PREMIS.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas sobre uso	<p>Todas estas subunidades de la unidad semántica son opcionales. Si se utiliza este contenedor debe estar presente por lo menos una de las subunidades <i>significantPropertiesValue</i> o <i>significantPropertiesExtension</i>. Si el contenedor <i>significantPropertiesExtension</i> tiene que asociarse con cualquier subunidad PREMIS dependiente de <i>significantProperties</i> se repite el contenedor <i>significantProperties</i>. Si hay que incluir extensiones para esquemas externos también es necesario repetir <i>significantProperties</i>.</p>		

Unidad semántica	1.5 objectCharacteristics		
Componentes semánticos	1.5.1 compositionLevel 1.5.2 fixity 1.5.3 size 1.5.4 format 1.5.5 creatingApplication 1.5.6 inhibitors 1.5.7 objectCharacteristicsExtension		
Definición	Propiedades técnicas aplicables a la totalidad o mayoría de los formatos.		
Justificación	Algunas propiedades técnicas importantes afectan a objetos en cualquier formato. La definición detallada de propiedades específicas de cada formato está fuera del ámbito de este Diccionario pero pueden incluirse en objectCharacteristicsExtension		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Obligatorio	Obligatorio
Notas de uso	<p>Las unidades semánticas de <i>objectCharacteristics</i> deben ser tratadas como un conjunto de información referida a un objeto simple con un <i>compositionLevel</i> simple. <i>objectCharacteristics</i> puede ser repetido cuando un objeto se haya creado aplicando más de una codificación como compresión y encriptado. En este caso cada repetición de <i>objectCharacteristics</i> tendría un nivel de composición <i>compositionLevel</i> superior. Cuando se aplica el encriptado el bloque <i>objectCharacteristics</i> debe incluir una unidad semántica inhibitors. Una cadena de bits embebida en un fichero puede tener características de objeto diferentes a las del fichero. Cuando estas características son relevantes para preservación deberían registrarse.</p> <p>Cuando un fichero simple es equivalente a una representación <i>objectCharacteristics</i> puede ser aplicado y asociado con la representación. En estos casos, la relación entre el fichero que incluye la representación y otros ficheros asociados puede ser expresada mediante relationshipSubType.</p>		

Unidad semántica	1.5.1 compositionLevel		
Componentes semánticos	Ninguno		
Definición	Indica si el objeto digital está sujeto a uno o varios procesos de decodificación.		
Justificación	Un fichero o una cadena de bits pueden estar comprimidos, cifrados...etc o empaquetados con otros ficheros o cadenas de bits en paquetes más grandes. Si hay que reconstruir el objeto u objetos originales es importante conocer el orden en el que se realizaron dichas acciones.		
Limitaciones	Números enteros		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Notas de mantenimiento	Por norma general compositionLevel debe ser identificado automáticamente por el sistema del Repositorio, generalmente durante el proceso de Ingest.		
Notas de creación /mantenimiento	<p>Un fichero o unas cadenas de bits pueden haber sufrido múltiples codificaciones que deben decodificarse en orden inverso (del más alto al más bajo. Por ejemplo, el fichero A debe ser comprimido para crear el fichero B, que está encriptado para crear el fichero C. Para recrear una copia del fichero base A se debe desencriptar el fichero C para crear el fichero B, después descomprimir el fichero B para crear el fichero A.</p> <p><i>compositionLevel</i> 0 indica que el objeto es un objeto base y no puede ser decodificado, pero un nivel uno o superior indica que pueden aplicarse más decodificaciones.</p> <p>La numeración va de mayor a menor (lo primero que se codifica es 0). 0 es un objeto base; 1-n son decodificaciones subsiguientes.</p> <p>Se utiliza 0 por defecto si solo hay un nivel de composición <i>compositionLevel</i>.</p> <p>Cuando hay varios ficheros o archivos que forman un paquete (ej, un fichero ZIP) los objetos ficheros individuales no son composition levels del objeto mayor. Deben ser tratados como objetos separados cada uno de ellos con sus unidades composition level. Por ejemplo: dos archivos encriptados en un zip que forman un fichero se describirían como tres objetos separados cada uno de ellos con sus propios metadatos. La localización de los dos objetos internos apuntaría al fichero ZIP, pero el fichero ZIP tendría un único nivel de composición (cero) y su formato sería "zip". Véase: Características y nivel de composición de los objetos: el modelo cebolla.</p>		

Unidad semántica	1.5.2 fixity		
Componentes semánticos	1.5.2.1 messageDigestAlgorithm 1.5.2.2 messageDigest 1.5.2.3 messageDigestOriginator		
Definición	Información utilizada para verificar si un objeto ha sido modificado sin autorización o de forma no documentada.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Notas de creación/mantenimiento	Calculado y registrado automáticamente por el repositorio.		
Notas de uso	<p>Un fixity check -test de fijeza- compara dos mensajes cifrados -message digest- calculados en momentos distantes en el tiempo. Si el resultado de la comprobación da el mismo valor, el objeto no ha sufrido alteraciones en el intermedio. Se recomienda utilizar dos o más tipos de mensajes cifrados calculados por procedimientos diferentes (Los términos message digest -mensaje cifrado- y checksum se utilizan indistintamente. Sin embargo, el término checksum es más apropiado para referirse al producto de una comprobación de redundancia cíclica mientras que mensaje cifrado -message digest- se refiere al resultado de una función hash y es al que se hace referencia aquí) La ejecución del fixity check y la fecha en que se realiza se registra como Acontecimiento. El resultado se registrará como <i>eventOutcome</i>. Por lo tanto sólo <i>messageDigestAlgorithm</i> y <i>messageDigest</i> tienen que ser registrados como <i>objectCharacteristics</i> para futuras comparaciones. Nivel de Representación: podría debatirse si una representación se compone de un sólo fichero o si todos los ficheros englobados en una representación se combinan en uno solo, (por ejemplo en un archivo ZIP) con lo que podría realizarse un fixity check sobre la representación. Sin embargo en ambos casos el fixity check se ejecuta sobre un fichero que es lo que coincide con la representación. Nivel de cadena de bits: los mensajes cifrados pueden calcularse para cadenas de bits aunque no sea habitual. Por ejemplo, el formato JPX, que es un formato JPEG2000 soporta la inclusión en los metadatos internos de mensajes cifrados MD5 o SHA-1 calculados sobre cualquiera de las cadenas de bytes del fichero. Véase Fijeza, integridad, autenticidad.</p>		

Unidad semántica	1.5.2.1 messageDigestAlgorithm		
Componentes semánticos	Ninguno		
Definición	Algoritmo concreto utilizado para elaborar el mensaje cifrado para el objeto digital.		
Justificación	Los valores deberían tomarse de un vocabulario controlado.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplos		MD5 Adler-32 HAVAL SHA-1 SHA-256 SHA-384 SHA-512 TIGER WHIRLPOOL	

Unidad semántica	1.5.2.2 messageDigest		
Componentes semánticos	Ninguno		
Definición	Resultado del algoritmo de cifrado.		
Justificación	Se almacena para que pueda compararse en comprobaciones futuras.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplos		7c9b35da4f2ebd436f 1cf88e5a39b3a257ed f4a22be3c955ac49da 2e2107b67a1924419 563	

Unidad semántica	1.5.2.3 messageDigestOriginator		
Componentes semánticos	Ninguno		
Definición	El agente que creó el mensaje cifrado original que se compara en un fixity check.		
Justificación	Un repositorio puede recibir ficheros con mensajes cifrados calculados por quien hace el proceso de carga de objetos digitales (ingest), su comprobación asegura que el fichero recibido es el mismo que el enviado. También puede recibir ficheros que no lleven mensajes cifrados por lo que hay que calcular los valores en el momento del proceso de carga de objetos digitales (ingest). Es útil conocer quien calculó el valor inicial del mensaje cifrado.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Ejemplos		DRS A0000978	
Notas de mantenimiento/ creación	Si el cálculo del mensaje cifrado inicial se considera un Acontecimiento se puede obtener esta información de un registro de acontecimiento.		
Notas de uso	El creador del mensaje cifrado puede representarse por una cadena (ej. "NRS" que se refiere al fichero) o apuntar a una descripción del agente (e.g., "A0000987" considerado como <i>agentIdentifierValue</i>).		

Unidad semántica	1.5.3 size		
Componentes semánticos	Ninguno		
Definición	El tamaño en bytes de un fichero o cadena de bits almacenado en el repositorio.		
Justificación	Permite asegurar que el tamaño del documento recuperado es efectivamente el mismo que cuando se guardó. Por otra parte el dato también es relevante a la hora de facturar el almacenamiento y calcular la provisión de espacio.		
Limitaciones	Número entero		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No aplicable	Aplicable	Aplicable
Repetible		No repetible	No repetible
Obligatoriedad		Opcional	Opcional
Ejemplo		2038937	
Notas mantenimiento	Obtenido directamente del repositorio.		
Notas de uso	Al definir esta unidad semántica como tamaño en bytes hace innecesario registrar una unidad de medida. Sin embargo para el intercambio de datos la unidad de medida debería acordarse entre las dos partes.		

Unidad semántica	1.5.4 format		
Componentes semánticos	1.5.4.1 formatDesignation 1.5.4.2 formatRegistry 1.5.4.3 formatNote		
Definición	Identificación del formato de un fichero o de una cadena de bits donde formato es la organización de la información digital según especificaciones predeterminadas.		
Justificación	Muchas de las actividades de preservación dependen del conocimiento detallado sobre el formato del objeto digital. Una identificación detallada del formato es esencial. La identificación proporcionada ya sea por su nombre o mediante un enlace a un registro debería ser suficiente para asociar el objeto con información adicional sobre el formato más detallada.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Obligatorio	Obligatorio
Notas de creación/mantenimiento	El formato de un fichero o cadena de bits debe ser consignado por el repositorio en el momento del proceso de carga de objetos digitales (ingest). Aunque el proveedor proporcione la información en los metadatos o esté en las extensiones de los ficheros se recomienda identificar el formato de forma independiente descomponiendo el fichero siempre que sea posible. Si no se puede identificar el formato en el momento del proceso de carga de objetos digitales (ingest) es válido registrar que es desconocido pero el repositorio debe esforzarse para identificar el formato aunque sea mediante intervención manual.		
Notas de uso	Una cadena de bits embebida en otro fichero puede tener características distintas a las del fichero mayor. Por ejemplo una cadena de bits en formato LaTeX puede embeberse en un fichero SGML o varias imágenes con espacios de color diferentes pueden estar embebidas en un TIFF. Debe registrarse la unidad <i>format</i> para cada de estos objetos. Cuando la cadena de bits pueda ser reconocida por el repositorio y el repositorio quiera tratar la cadena de forma diferente a la del fichero que las agrupa, <i>format</i> puede registrarse para las cadenas embebidas. Aunque esta unidad semántica es obligatoria las dos subunidades son opcionales. Por lo menos una de las subunidades (<i>formatDesignation</i> o <i>formatRegistry</i>) debe estar presente si se incluye este contenedor. Si hay que repetir una de las subunidades (<i>formatDesignation</i> o <i>formatRegistry</i>) se repite todo el contenedor <i>format</i> . Esto contempla la asociación de la designación de formato con un conjunto determinado de información sobre formatos. Por ejemplo, si el formato concreto no puede ser determinado y se registran dos designaciones <i>format</i> cada una de ellas se da en un		

	contenedor <i>format</i> separado. El contenedor <i>format</i> también puede repetirse para entradas múltiples desde el registro de formatos. Véase Información de formato.
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Unidad semántica	1.5.4.1 formatDesignation		
Componentes semánticos	1.5.4.1.1 formatName 1.5.4.1.2 formatVersion		
Definición	Identificación del formato del objeto digital.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Notas de uso	<p>Se exige <i>formatDesignation</i> o por lo menos alguna parte de <i>formatRegistry</i>. Pueden incluirse las dos. Debe registrarse el formato más específico (o un perfil de formato). Un repositorio (o un registro de formatos) puede desear utilizar nombres de formatos en varias partes (ej. "TIFF_GeoTIFF " o "WAVE_MPEG_BWF") para alcanzar esta especificidad.</p>		

Unidad semántica	1.5.4.1.1 formatName		
Componentes semánticos	Ninguno		
Definición	Nombre del formato del fichero o de la cadena de bits.		
Limitaciones	El valor debería tomarse de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplos		Text/sgml image/tiff/geotiff Adobe PDF DES PGP base64 unknown	LaTeX
Notas de uso	Puede apuntarse "desconocido" en caso de formatos no identificados.		

Unidad semántica	1.5.4.1.2 formatVersion		
Componentes semánticos	Ninguno		
Definición	La versión del formato mencionado en <i>formatName</i> .		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Ejemplos		6.0 2003	
Notas de uso	Si hay versiones del formato debería registrarse <i>formatVersion</i> . Puede utilizarse una designación numérica o cronológica.		

Unidad semántica	1.5.4.2 formatRegistry		
Componentes semánticos	1.5.4.2.1 formatRegistryName 1.5.4.2.2 formatRegistryKey 1.5.4.2.3 formatRegistryRole		
Definición	Identifica o proporciona información más detallada sobre el formato haciendo referencia a una entrada de un registro de formatos.		
Justificación	Los registros centralizados de formatos proporcionan una manera excelente de consignar información detallada sobre formatos.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Notas mantenimiento	El grupo de trabajo de PREMIS supone que se implementará un registro de formatos digitales como apoyo a la preservación digital global. Como iniciativa ejemplar destaca (GDFR) http://hul.harvard.edu/qdfr/documents.html#data		

Unidad semántica	1.5.4.2.1 formatRegistryName		
Componentes semánticos	Ninguno		
Definición	Designación que identifica el registro de formatos referenciado.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplos		PRONOM www.nationalarchives.gov.uk/pronom Representation Information Registry Repository	FRED
Notas de uso	Puede ser un nombre formal, un nombre utilizado internamente o un URI (uniform resource identifier)		

Unidad semántica	1.5.4.2.2 formatRegistryKey		
Componentes semánticos	Ninguno		
Definición	Clave única utilizada para referirse a una entrada de un registro de formatos sobre este formato.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplos		info:gdf/fred/f/tiff TIFF/6.0	

Unidad semántica	1.5.4.2.3 formatRegistryRole		
Componentes semánticos	Ninguno		
Definición	Se puede definir el mismo <i>format</i> en diferentes registros con diferentes propósitos. Por ejemplo un registro puede dar especificaciones detalladas del formato mientras que otro registro da información sobre el perfil. Si se consignan múltiples registros esta unidad semántica puede utilizarse para diferenciarlos.		
Limitaciones	El valor debe ser tomado de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Ejemplos		Especificación Perfil de validación	

Unidad semántica	1.5.4.3 formatNote		
Componentes semánticos	Ninguno		
Definición	Información adicional referente al formato.		
Justificación	En ocasiones, la información cualificada de un formato concreto puede ser necesaria para su posterior recuperación.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Ejemplo		Identificación provisional Encontradas múltiples identificaciones de formato	
Notas de uso	Puede contener texto libre, un enlace externo o un valor de un vocabulario controlado.		

Unidad semántica	1.5.5 creatingApplication		
Componentes semánticos	1.5.5.1 creatingApplicationName 1.5.5.2 creatingApplicationVersion 1.5.5.3 dateCreatedByApplication 1.5.5.4 creatingApplicationExtension		
Definición	Información referente a la aplicación con la que se ha creado el objeto.		
Justificación	Información referente a las aplicaciones con las que se crean los objetos digitales. Incluye la versión y la fecha de creación del fichero. Es útil, por ejemplo en los casos conocidos en que ciertas versiones de programas causan errores identificados durante la conversión a su nueva versión. También es útil para determinar qué prestación de software podrá utilizarse para visualizar el objeto digital. Por ejemplo, si sabemos que ha sido el programa Distiller, con el que hemos creado un fichero PDF, sabremos a posteriori que (entre otros programas) podemos utilizar el Adobe Reader para leer el documento.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Notas de creación/mantenimiento	Si el objeto ha sido creado por la institución, la obtención de este dato es sencillo. Si el objeto se ha creado fuera, es posible que esta información pueda ser proporcionada por el depositante. No obstante también puede extraerse de los ficheros en sí, ya que el nombre de la aplicación con que se ha creado un fichero a menudo está integrado en el fichero.		
Notas de uso	Esta unidad semántica se refiere tanto a objetos creados fuera del repositorio e incluidos posteriormente como a objetos creados por el repositorio, por ejemplo mediante migraciones. El contenedor <i>creatingApplication</i> es repetible si a su vez más de una aplicación ha procesado el objeto. Por ejemplo, un fichero puede ser creado con Microsoft Word y convertido después en PDF con Adobe Acrobat. Hay que registrar detalles tanto de Word como de Acrobat. Sin embargo, si los dos ficheros se almacenan en el repositorio cada fichero debe ser descrito en su totalidad como entidad Objeto y relacionados utilizando información de relación mediante <i>relationshipType</i> "derivado." También podría repetirse para registrar la aplicación que lo creó antes de que se incluyera así como la aplicación creadora utilizada como parte del proceso de carga de objetos digitales (ingest), por ejemplo un fichero HTML se creó antes del ingest utilizando Dreamweaver y posteriormente Heritrix capturó unos pantallazos de los ficheros como parte del ingest. La cantidad de información necesaria que se da para <i>creatingApplication</i> es mínima. Se proporciona extensibilidad para más detalles. Es preferible tener un registro de esta información similar a la de los registros de formatos o de entornos frente a hacer que cada repositorio registre localmente esta información.		

Unidad semántica	1.5.5.1 creatingApplicationName		
Componentes semánticos	Ninguno		
Definición	Nombre del programa con el cual se ha creado el objeto digital.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Ejemplo		MsWord	
Notas de uso	<p><i>creatingApplication</i> se refiere a la aplicación que creó el objeto en el formato original no a la aplicación que creó la copia de archivo. Por ejemplo, si un documento se creó con Microsoft Word y posteriormente se hizo una copia para almacenarlo por un proceso de carga de objetos digitales (ingest), <i>creatingApplication</i> es Word, no el programa de Ingest.</p>		

Unidad semántica	1.5.5.2 creatingApplicationVersion		
Componentes semánticos	Ninguno		
Definición	Versión del programa con el cual se ha creado el objeto digital.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Ejemplo		2000	1.4

Unidad semántica	1.5.5.3 dateCreatedByApplication		
Componentes semánticos	Ninguno		
Definición	Fecha y hora de creación del objeto puede ser aproximada si no se conoce la exacta.		
Limitaciones	Este valor debería adoptar una forma estructurada para facilitar el proceso automatizado de los datos. Se recomienda utilizar convenciones normalizadas que faciliten el intercambio de metadatos conforme a PREMIS, por ejemplo las utilizadas en el esquema PREMIS.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Ejemplo		2000-12-01 20030223T151047	
Nota de uso	<p>Se recomienda el uso de la fecha más precisa que se conozca. Esta fecha es la de creación del objeto original, no se refiere a la fecha en la que se hizo alguna copia. Por ejemplo, si un fichero se creó en Microsoft Word en el 2001 y se hacen dos copias del mismo en el 2003, el valor de <i>dateCreatedByApplication</i> de los tres ficheros es de 2001.</p> <p>La fecha de almacenamiento de un fichero puede registrarse como Evento. Si el objeto incluye fechas internas de creación o de modificación debe utilizarse la fecha de modificación como <i>dateCreatedByApplication</i>.</p> <p>Si el objeto es capturado por un Web harvester se puede registrar la fecha de captura.</p>		

Unidad semántica	1.5.5.4 creatingApplicationExtension		
Componentes semánticos	Definidos externamente		
Definición	Información de la aplicación con la que se ha creado el objeto digital obtenida de unidades semánticas externas a las establecidas en PREMIS.		
Justificación	Sustituir o ampliar unidades las semánticas definidas en PREMIS.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
	<p>La extensibilidad se proporciona para un mayor nivel de detalle o para el uso de unidades semánticas definidas fuera del Diccionario. Se pueden incluir tanto unidades semánticas locales como metadatos procedentes de otro esquema de metadatos bien en adición bien en sustitución de las unidades semánticas definidas en el Diccionario PREMIS.</p> <p>Cuando se utilice un esquema de extensión, se debe proporcionar una referencia al esquema utilizado. Para obtener más información relativa a la "extensibilidad" véase Capacidad de Extensión.</p> <p>Si el contenedor <i>creatingApplicationExtension</i> tiene que estar necesariamente asociado con alguna de las subunidades PREMIS dependientes de <i>creatingApplication</i>, se repite el contenedor. Si es necesario recurrir a la <i>extensibilidad</i> mediante esquemas de metadatos externos, el contenedor <i>creatingApplication</i> también debe repetirse.</p>		

Unidad semántica	1.5.6 inhibitors		
Componentes semánticos	1.5.6.1 inhibitorType 1.5.6.2 inhibitorTarget 1.5.6.3 inhibitorKey		
Definición	Características de los objetos destinados a impedir el acceso, el uso, o la migración.		
Justificación	Se debe indicar si un fichero está encriptado, y también la naturaleza del encriptado y la manera de desencriptarlo.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Notas mantenimiento	Por lo general los Inhibidores de acceso están presentes en objetos digitales que han pasado a formar parte de un repositorio por un proceso de carga de objetos digitales (ingest). No suele ser posible identificar si un objeto está o no encriptado a no ser que se analice y se descomponga. Por lo tanto, se debería proporcionar siempre que sea posible la información sobre los inhibidores en forma de metadatos cuando el objeto se remita al repositorio.		
Notas de uso	Algunos formatos permiten que las cadenas de bits estén encriptadas, formatos como PDF utilizan contraseñas para regular el acceso al contenido o a alguna función concreta. Aunque normalmente esto se ejecuta a nivel de cadena de bits, a efectos de preservación se gestiona eficazmente a nivel de fichero, o sea que las contraseñas no se registrarán para cadenas de bits accesibles por separado. Se requerirá mayor nivel de detalle para algunos tipos de claves. Si la información sobre la clave es idéntica a la de las firmas digitales hay que utilizar esas unidades semánticas.		

Unidad semántica	1.5.6.1 inhibitorType		
Componentes semánticos	Ninguno		
Definición	Tipo de Inhibidor de acceso utilizado.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplo		DES PGP Blowfish Password protection	
Notas de uso	El encriptado y las contraseñas son los inhibidores más utilizados. Cuando se utiliza el encriptado hay que indicar el tipo específico, es decir se anota "DES" (Data Encryption Standard) y no "encryption".		

Unidad semántica	1.5.6.2 inhibitorTarget		
Componentes semánticos	Ninguno		
Definición	Contenido o función para el cual se aplica el inhibidor de acceso.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Ejemplo		Todo el contenido Función: play Función: Impresión	
Notas de uso	Si no se cumplimenta esta unidad se considera que el inhibidor afecta a todo el objeto digital.		

Unidad semántica	1.5.6.3 inhibitorKey		
Componentes semánticos	Ninguno		
Definición	La clave de acceso o descriptado.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Ejemplos		[DES decryption key]	
Notas de uso	Debe proporcionarse la clave si se conoce. Sin embargo no se aconseja almacenar <i>inhibitorKey</i> en texto plano en una base de datos no segura.		

Unidad semántica	1.5.7 objectCharacteristicsExtension		
Componentes semánticos	Definido externamente		
Definición	Contenedor para unidades semánticas no definidas en el Diccionario de Datos PREMIS.		
Justificación	Sustituir o ampliar unidades semánticas definidas en PREMIS.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Notas de uso	<p>La extensibilidad se proporciona para un mayor nivel de detalle o para el uso de unidades semánticas definidas fuera del Diccionario. Se pueden incluir tanto unidades semánticas locales como metadatos procedentes de otro esquema de metadatos bien en adición bien en sustitución de las unidades semánticas definidas en el Diccionario PREMIS.</p> <p>Cuando se utilice un esquema de extensión, se debe proporcionar una referenciar al esquema utilizado. Para obtener más información relativa a la "extensibilidad" véase Capacidad de extensión.</p> <p>Si el contenedor <i>objectCharacteristicsExtension</i> tiene que estar necesariamente asociado con alguna de las subunidades <i>objectCharacteristics</i>, se repite el contenedor. Si es necesario recurrir a la <i>extensibilidad</i> mediante esquemas de metadatos externos, el contenedor <i>objectCharacteristics</i> también debe repetirse.</p>		

Unidad semántica	1.6 originalName		
Componentes semánticos	Ninguno		
Definición	El nombre del objeto digital tal cual es entregado o harvestado por el repositorio, antes de que el repositorio lo renombre según sus normas.		
Justificación	El nombre del objeto digital que se utilice en el repositorio de preservación puede ser desconocido fuera del repositorio. Un depositante en ocasiones solicita un fichero por su nombre original. Por otra parte el repositorio puede tener que reconstruir enlaces internos de difusión y para ello es necesario el nombre original del mismo. En el caso de las representaciones, debe existir un directorio de los nombres.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	No aplicable
Repetible	No repetible	No repetible	
Obligatoriedad	Opcional	Opcional	
Ejemplo		.pdf	
Notas de creación / mantenimiento	Esta información siempre será proporcionada por el proveedor o por la aplicación que efectúa la recogida de la información. El repositorio decide cual es la porción del nombre original que quiere conservar.		
Notas de uso	Es el nombre que recibe el objeto en el SIP (Submission Information Package) Este objeto puede tener nombres diferentes en otros contextos. Cuando se intercambia el contenido entre dos repositorios es importante que el receptor conozca y registre el nombre que tenía la representación en el repositorio de origen. En el caso de las representaciones será el nombre de un directorio.		

Unidad semántica	1.7 storage		
Componentes semánticos	1.7.1 contentLocation 1.7.2 storageMedium		
Definición	Información sobre cómo y dónde se almacena un fichero en un repositorio.		
Justificación	Es necesario asociar el contenido de <i>contentLocation</i> y de <i>storageMedium</i> .		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Obligatorio	Obligatorio
Notas de uso	<p>Normalmente hay un único <i>contentLocation</i> y <i>storageMedium</i> por objeto, ya que un objeto almacenado en un lugar diferente sería considerado un objeto diferente. Sin embargo el almacenamiento de un documento puede repetirse si hay dos o más ejemplares que son idénticos. En este caso se considera una unidad excepto para el medio en el que están almacenados. Debe tener un <i>objectIdentifier</i> único y se gestionará como un único objeto por el repositorio.</p> <p>Aunque esta unidad semántica es obligatoria, sus dos componentes semánticos son opcionales. Por lo menos una de las subunidades (i.e <i>contentLocation</i> o <i>storageMedium</i>) debe estar presente.</p>		

Unidad semántica	1.7.1 contentLocation		
Componentes semánticos	1.7.1.1 contentLocationType 1.7.1.2 contentLocationValue		
Definición	Información necesaria para recuperar un fichero o para acceder a una cadena de bits del repositorio		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Notas de mantenimiento	Un repositorio de preservación nunca debería contener objetos digitales que no estuvieran debidamente identificados y localizados. Por tanto el Grupo de trabajo PREMIS asume que debe de ser el propio repositorio digital el responsable de asignar <i>contentLocation</i> de manera automática.		
Notas de uso	Si el repositorio utiliza <i>objectIdentifier</i> para la recuperación de datos, <i>contentLocation</i> está implícito y no es necesario y por lo tanto es opcional.		

Unidad semántica	1.7.1.1 contentLocationType		
Componentes semánticos	Ninguno		
Definición	Forma de referenciar la localización del contenido.		
Justificación	Para comprender el significado del valor, es necesario conocer previamente el esquema de localización que se va a utilizar.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplo		URI hdl NTFS EXT3	byte offset

Unidad semántica	1.7.1.2 contentLocationValue		
Componentes semánticos	Ninguno		
Definición	Localización del contenido en el repositorio.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplo		URI: http://wwasearch.loc.gov/107th/200212107035/http://house.gov/langevin/ hdl: loc.pnp/cph.3b34 NTFS: c:\apache2\htdocs\index.html EXT3: home/web/public_html/index.html	64 [offset from start of file c:\apache2\htdocs\image\logo.gif]
Notas de uso	Puede tratarse de una ruta completa o de la información utilizada por un sistema de resolución (ej. handle) o de la información nativa utilizada por un gestor de almacenamiento. Será probablemente el punto de referencia y de partida de la cadena de bits. El repositorio decide el grado de detalle que tiene que registrar.		

Unidad semántica	1.7.2 storageMedium		
Componentes semánticos	Ninguno		
Definición	Soporte físico de almacenamiento del objeto digital.		
Justificación	Se debe conocer cuál es el soporte de almacenamiento con el fin de saber con posterioridad como y cuando hacer migraciones.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No aplicable	Aplicable	Aplicable
Repetible		No repetible	No repetible
Obligatoriedad		Opcional	Opcional
Ejemplo		Cinta magnética DVD	Cinta magnética DVD
Notas de uso	En algunos casos los gestores de almacenamiento no envían este dato al repositorio pero se entiende que el repositorio tiene que conocer este dato para gestionar el problema de la obsolescencia tecnológica. Es posible que el valor no sea el soporte específico sino el sistema que maneja el soporte, por ejemplo Tivoli Storage Manager (TSM). Conocer el soporte de almacenamiento es un requisito interno para poder lanzar las acciones de preservación pero es opcional ya que no se utiliza para intercambiar datos.		

Unidad semántica	1.8 environment		
Componentes semánticos	1.8.1 environmentCharacteristic 1.8.2 environmentPurpose 1.8.3 environmentNote 1.8.4 dependency 1.8.5 software 1.8.6 hardware 1.8.7 environmentExtension		
Definición	Combinación de información referente al hardware y software de apoyo a la utilización del objeto digital.		
Justificación	Conocer el contexto permite al usuario interactuar y utilizar el contenido del objeto digital. Separar los contenidos digitales de su contexto original puede tener consecuencias nefastas para su posterior utilización.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas mantenimiento	Se puede omitir esta información cuando el repositorio únicamente pretenda preservar el acceso a nivel de bits del objeto digital. Es preferible contar con un registro de información sobre entornos similar a la propuesta para formatos antes que optar porque cada repositorio registre esta información de forma local. Véase Entorno .		
Notas de uso	Estas subunidades son opcionales. Si se incluye este contenedor debe aparecer por lo menos una de ellas (i.e. <i>environmentNote</i> , <i>dependency</i> , <i>software</i> , <i>hardware</i> , y/o <i>environmentExtension</i>).		

Unidad semántica	1.8.1 environmentCharacteristic		
Componentes semánticos	Ninguno		
Definición	Evaluación del nivel que alcanza el entorno descrito en la consecución de su objetivo.		
Justificación	Este elemento puede ayudar a distinguir entre varios entornos.		
Limitaciones	El valor debe tomarse de un vocabulario controlado		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas mantenimiento	El valor puede proceder bien del depositante, bien ser asignado por el repositorio durante el proceso de carga de objetos digitales (ingest). No obstante debe tenerse en cuenta que el significado de cada valor debe de ser único tanto si es asignado externamente como por el propio repositorio.		
Notas de uso	<p>Sugerencia de valores a tener en cuenta:</p> <p>Unspecified: No se dispone información referente al entorno o contexto de Software/Hardware necesario.</p> <p>Minimum: Se sabe que con los mínimos recursos de Hardware/Software el objeto se visualiza.</p> <p>Recommended: Se recomienda la utilización de un entorno específico, pues se sabe que es el más apropiado para la visualización de un objeto digital.</p> <p>Known to work: Supone que el objeto es soportado por el entorno descrito. Pero se desconoce si este valor es Mínimo o es el verdaderamente Recomendado.</p>		

Unidad semántica	1.8.2 environmentPurpose		
Componentes semánticos	Ninguno		
Definición	La finalidad de cada una de las características del entorno.		
Justificación	Por ejemplo, se pueden necesitar diferentes entornos a la hora de manipular un objeto digital. Uno para editarlo o modificarlo y otro diferente para visualizarlo.		
Limitaciones	El valor debe tomarse de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas de creación / mantenimiento	Este valor puede proceder bien del depositante bien ser asignado por el repositorio durante el proceso de carga de objetos digitales (ingest). Si la información del entorno de hardware y software se obtiene de un registro de entornos también el dato <i>environmentCharacteristic</i> puede obtenerse también del registro Hay que destacar que el criterio para "recommended" puede ser diferente en cada repositorio.		
Notas de uso	Sugerencia de valores a tener en cuenta: Edit Print Transform Render		

Unidad semántica	1.8.3 environmentNote		
Componentes semánticos	Ninguno		
Definición	Información adicional sobre las características del entorno.		
Justificación	En ocasiones es necesario incorporar información adicional respecto al entorno en una nota de texto plano.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo		Este documento en PDF no necesita un programa específico de lectura, sirve cualquier lector estándar de PDF	
Notas de uso	Esta nota puede utilizarse para registrar el contexto de la información sobre el entorno. Por ejemplo, si un fichero puede obtenerse por medio de una aplicación cliente para PC o por un navegador con plug-in esta nota puede utilizarse para identificar qué situación se aplica. Esta nota no debería utilizarse para una descripción textual sobre el entorno que se registra de forma más rigurosa en otro lugar.		

Unidad semántica	1.8.4 dependency		
Componentes semánticos	1.8.4.1 dependencyName 1.8.4.2 dependencyIdentifier		
Definición	Información no del software, sino de un componente o fichero asociado y que es necesaria para el correcto uso de la representación o el fichero, por ejemplo, el esquema de la estructura del documento, o una DTD.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas mantenimiento	Práctica recomendada en repositorios que almacenen documentos de los cuales dependen otros objetos digitales. Por ejemplo, un fichero de marcado a menudo contiene enlaces a otros objetos que le dan la estructura que requiere, como una DTD o un XML Schema. Si se identifican en esta unidad, a la hora de visualizar el documento, estos objetos requeridos se identifican mediante el vínculo y se descargan por el repositorio.		
Notas de uso	<p>Esta unidad semántica se utiliza para objetos adicionales necesarios para recuperar un fichero o una representación no se utiliza para software o hardware. También se puede utilizar para un componente no ejecutable del objeto como un tipo de letra o una hoja de estilo. Véase swDependency para elementos que requieran el software.</p> <p>Esta unidad semántica no se utiliza para especificar relaciones estructurales como pueden ser las que hay entre un objeto <i>padre/principal</i> y los <i>child</i> (por ejemplo un artículo científico y las figuras que le acompañan). Esta información se registra en la unidad <i>relationshipType</i> como "structural".</p> <p>Los objetos tratados pueden estar dentro o fuera del repositorio.</p>		

Unidad semántica	1.8.4.1 dependencyName		
Componentes semánticos	Ninguno		
Definición	Designación para un componente o un fichero asociados necesarios para la representación o para el fichero.		
Justificación	El nombre del objeto puede que no sea evidente únicamente a partir del valor <i>dependencyIdentifier</i> .		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo		Elemento Adicional	

Unidad semántica	1.8.4.2 dependencyIdentifier		
Componentes semánticos	1.8.4.2.1 dependencyIdentifierType 1.8.4.2.2 dependencyIdentifierValue		
Definición	Denominación única dada para identificar un recurso dependiente.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas de uso	<i>dependencyIdentifier</i> debe ser único en el repositorio.		

Unidad semántica	1.8.4.2.1 dependencyIdentifierType		
Componentes semánticos	Ninguno		
Definición	Denominación del dominio en el que el identificador del recurso dependiente es único.		
Limitaciones	El valor debería tomarse de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Notas de uso	Un repositorio de preservación necesita conocer tanto el tipo de identificador del objeto como su valor. Cuando el propio valor contenga el tipo de identificador (ej: "oai:lib.uchicago.edu:1"), no hay que registrar el tipo de identificador. Igualmente si el repositorio utiliza solo un tipo de identificador, el tipo se sobreentiende y no hay que registrarlo.		

Unidad semántica	1.8.4.2.2 dependencyIdentifierValue		
Componentes semánticos	Ninguno		
Definición	Valor de <i>dependencyIdentifier</i>		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplos		http://www.tei-c.org/P4X/DTD/teicorp2.dtd	

Unidad semántica	1.8.5 software		
Componentes semánticos	1.8.5.1 swName 1.8.5.2 swVersion 1.8.5.3 swType 1.8.5.4 swOtherInformation 1.8.5.5 swDependency		
Definición	Software necesario para visualizar o utilizar el objeto digital.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas de creación/mantenimiento	<p>Si se registra explícitamente, puede aplicarse a diferentes entornos, por ejemplo, un objeto concreto, como un fichero PDF, puede visualizarse mediante diferentes versiones de diferentes aplicaciones que funcionan con diferentes sistemas operativos y sus diferentes versiones. No es necesario registrarlo todo aunque si haya que consignar por lo menos un entorno de software. Cada repositorio tendrá que decidir qué entorno registrar. También hay que tener en cuenta que lo que se ve como un solo programa puede tener dependencias, como utilidades del sistema o bibliotecas de rutinas que a su vez tengan también sus dependencias. Al igual que en el entorno los metadatos serán gestionados de forma más eficaz si se cuenta con un registro interno o externo. Al faltar un mecanismo global los repositorios pueden verse forzados a desarrollar sus propios "registros" para relacionar formatos y entornos de software.</p>		

Unidad semántica	1.8.5.1 swName		
Componentes semánticos	Ninguno		
Definición	Nombre del software y empresa del software		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplos	Sybase	Adobe Photoshop Adobe Acrobat Reader	
Notas mantenimiento	El hecho de incluir el nombre de la empresa del software puede ayudar a identificar unívocamente el producto por ejemplo, es mejor utilizar el término <i>Adobe Photoshop</i> que simplemente <i>PhotoShop</i> .		

Unidad semántica	1.8.5.2 swVersion		
Componentes semánticos	Ninguno		
Definición	Versión o versiones del software referenciado en <i>swName</i> .		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo		>=2.2.0 6.0 2000	
Notas de uso	En caso de no existir una versión formalizada del software se puede utilizar la fecha del mismo.		

Unidad semántica	1.8.5.3 swType		
Componentes semánticos	Ninguno		
Definición	Clase o categoría de <i>software</i>		
Limitaciones	El valor debe tomarse de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Notas de uso	<p>Renderer: Aplicación que visualiza/ejecuta el formato. Por ejemplo image viewer, video player, Java virtual machine.</p> <p>Ancillary: Cuando es necesario un software auxiliar de apoyo. Por ejemplo plug-ins, rutinas de compression/decompression.</p> <p>operatingSystem: Software que sustenta la ejecución de la aplicación, los procesos, la gestión de la memoria, etc.</p> <p>Driver: Software cuya función es establecer la comunicación entre el software y el hardware.</p>		

Unidad semántica	1.8.5.4 swOtherInformation		
Componentes semánticos	Ninguno		
Definición	Requisitos adicionales o instrucciones referidas al software registrado en <i>swName</i> .		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo		Instalar Acroread (Adobe Acrobat): Para ello primero copiar nppdf.so (el plug-in) en el directorio de plug-in de Mozilla	
Notas de uso	Puede tratarse de un identificador permanente de confianza o de un URI (Uniform Resource Identifier) que enlace con documentación sobre el software.		

Unidad semántica	1.8.5.5 swDependency		
Componentes semánticos	Ninguno		
Definición	Nombre y si procede la versión de cualquier componente de software que necesite el software que se menciona en <i>swName</i> para utilizar este objeto		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo		GNU gcc >= 2.7.2	
Notas de uso	El valor se debe construir de forma acorde con la construcción de <i>swName</i> y de <i>swVersion</i> . Esta unidad semántica identifica el software que se necesita a través de lo que se registra en <i>swName</i> , Por ejemplo un script en Perl que depende de un módulo en Perl. En este caso concreto el script en Perl se registrará en <i>swName</i> , y el módulo en Perl en <i>swDependency</i> dentro de un contenedor de software.		

Unidad semántica	1.8.6 hardware		
Componentes semánticos	1.8.6.1 hwName 1.8.6.2 hwType 1.8.6.3 hwOtherInformation		
Definición	Componentes de hardware necesarios para que el software referenciado en <i>swName</i> funcione o el usuario de dicho software.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas mantenimiento	<p>Es muy complicado aportar información detallada del hardware debido a que existen innumerables combinaciones de equipos de CPU, memorias, drivers, etc. No es necesario registrarlo todo aunque si haya que consignar por lo menos un entorno de hardware. Cada repositorio tendrá que decidir qué entorno registrar.</p> <p>Debido a la dificultad para registrar esta información en su totalidad, lo ideal sería que existieran registros centralizados de información sobre entornos. En muchos casos el entorno de un fichero está directamente asociado al formato, lo que posibilita la búsqueda por formato. Al no existir un mecanismo global los repositorios pueden verse forzados a desarrollar "registros" locales para enlazar formatos con <i>hwEnvironment</i>.</p>		

Unidad semántica	1.8.6.1 hwName		
Componentes semánticos	Ninguno		
Definición	Fabricante, modelo, y versión (si procede) del hardware		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo		Intel Pentium III 1 GB DRAM	
Notas mantenimiento	Incluir el nombre del fabricante cuando esto ayude a identificar o el producto. Incluir la versión de la programación en firme u otros componentes si es pertinente.		

Unidad semántica	1.8.6.2 hwType		
Componentes semánticos	Ninguno		
Definición	Clase o categoría del hardware.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Notas mantenimiento	Posibles valores: procesador, memoria, dispositivo de entrada / salida, almacenamiento.		

Unidad semántica	1.8.6.3 hwOtherInformation		
Componentes semánticos	Ninguno		
Definición	Información complementaria o instrucciones que hacen referencia al hardware mencionado en <i>hwName</i> .		
Justificación	Es posible que haya que documentar la cantidad de recursos que se necesitan (memoria, almacenamiento, velocidad del procesador, etc.) para el hardware. Además pueden necesitarse instrucciones más detalladas para instalar y/o hacer funcionar el hardware.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo	32 MB como mínimo	32 MB como mínimo La cantidad de memoria RAM que se requiere para el Apache se desconoce	
Notas de uso	Se puede utilizar un Identificador o URI que enlace con la documentación sobre el hardware.		

Unidad semántica	1.8.7 environmentExtension		
Componentes semánticos	Definidos externamente		
Definición	Contenedor para incluir unidades semánticas definidas fuera del Diccionario de Datos PREMIS.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas de uso	<p>La extensibilidad se proporciona para un mayor nivel de detalle o para el uso de unidades semánticas definidas fuera del Diccionario. Se pueden incluir tanto unidades semánticas locales como metadatos procedentes de otro esquema de metadatos bien en adición bien en sustitución de las unidades semánticas definidas en el Diccionario PREMIS.</p> <p>Cuando se utilice un esquema de extensión, se debe proporcionar una referencia al esquema utilizado. Para obtener más información relativa a la "extensibilidad" véase capacidad de extensión.</p> <p>Si el contenedor <i>environmentExtension</i> tiene que estar necesariamente asociado con alguna de las subunidades PREMIS de entorno, se repite el contenedor <i>environment</i>. Si es necesario recurrir a la <i>extensibilidad</i> mediante esquemas de metadatos externos, el contenedor <i>environment</i> también debe repetirse.</p>		

Unidad semántica	1.9 signatureInformation		
Componentes semánticos	1.9.1 signature 1.9.2 signatureInformationExtension		
Definición	Un contenedor para información sobre la firma digital que puede estar desarrollada tanto en el Diccionario de Datos PREMIS como externamente. Se utiliza para autenticar la firma y/o la información.		
Justificación	Un repositorio puede tener la política de generar firmas digitales durante el proceso de carga de objetos digitales (ingest), o puede necesitar almacenar las firmas digitales y validarlas posteriormente.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Notas mantenimiento	Se puede utilizar tanto <i>signature</i> como <i>signatureInformationExtension</i> . Se recomienda el uso de <i>signatureInformationExtension</i> con el esquema definido en W3C's <i>XML-Signature Syntax and Processing</i> (http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/). Véase la información sobre firmas digitales para más información sobre el uso de unidades semánticas bien sean definidas en PREMIS bien sean definidas externamente.		

Unidad semántica	1.9.1 signature		
Componentes semánticos	1.9.1.1 signatureEncoding 1.9.1.2 signer 1.9.1.3 signatureMethod 1.9.1.4 signatureValue 1.9.1.5 signatureValidationRules 1.9.1.6 signatureProperties 1.9.1.7 keyInformation		
Definición	Información necesaria en caso de utilizar una firma digital para autenticar al firmante de un objeto y / o la información contenida en el mismo.		
Justificación	Un repositorio puede tener la política de generar firmas digitales durante el proceso de carga de objetos digitales (ingest) o puede necesitar almacenar las firmas digitales y validarlas posteriormente. o bien tomar la decisión de almacenar los ficheros y con posterioridad validarlos con firmas digitales.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Notas de uso	Varios de los componentes semánticos de <i>signatureInformation</i> están tomados de W3C's <i>XML-Signature Syntax and Processing</i> , véase http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/ . Véase también la información sobre firmas digitales .		

Unidad semántica	1.9.1.1 signatureEncoding		
Componentes semánticos	Ninguno		
Definición	Código utilizado para cumplimentar los valores de <i>signatureValue</i> , <i>keyInformation</i> .		
Justificación	Es necesario conocer el código utilizado para interpretar correctamente ambos valores.		
Limitaciones	El valor debe proceder de un vocabulario controlado		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplo		Base64 Ds:CryptoBinary	

Unidad semántica	1.9.1.2 signer		
Componentes semánticos	Ninguno		
Definición	Persona, institución o autoridad responsable de generar la firma digital.		
Justificación	Este dato también puede ser incluido en <i>keyInformation</i> pero es más conveniente que aparezca aquí.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Ejemplo	En caso de que el firmante sea un <i>Agent</i> identificado en el repositorio, se puede utilizar aquí el valor <i>agentIdentifier</i> .		

Unidad semántica	1.9.1.3 signatureMethod		
Componentes semánticos	Ninguno		
Definición	Nombre o designación del cifrado o algoritmo utilizado para generar las firmas.		
Justificación	El mismo tipo de algoritmo es el que debe de ser utilizado para validar la firma.		
Limitaciones	El valor debe proceder de un lenguaje controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplo		DSA-SHA1 RSA-SHA1	
Notas de uso	Se recomienda codificar primero el algoritmo de cifrado, seguido de un guión y seguido de un algoritmo hash (mensaje cifrado).		

Unidad semántica	1.9.1.4 signatureValue		
Componentes semánticos	Ninguno		
Definición	La firma digital, un valor generado mediante la aplicación de un código privado al mensaje cifrado.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Ejemplo		juS5RhJ884qoFR 8fVXd/rbrSDVGn 40CapgB7qeQiT +rr0NekEQ6BHh UA8dT3+BCTBU QI0dBjImI9lwzEN XvS83zRECjzXb MRTUtVZiPZG2p qKPnL2YU3A964 5UCjTXU+jgFum v7k78hieAGDzNc i+PQ9KRmm//icT 7JaYztgt4=	

Unidad semántica	1.9.1.5 signatureValidationRules		
Componentes semánticos	Ninguno		
Definición	Operaciones necesarias para validar la firma digital.		
Justificación	No se debe asumir que en el futuro seamos capaces de interpretar o conocer el procedimiento de validación de todas las firmas digitales a no ser que se documenten.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Obligatorio	Obligatorio
Notas de uso	Puede incluir el método de normalización utilizado antes de calcular el mensaje cifrado –message digest- en caso de que el objeto fuera normalizado antes de la firma. Este valor también puede ser un enlace a la documentación.		

Unidad semántica	1.9.1.6 signatureProperties		
Componentes semánticos	Ninguno		
Definición	Información adicional referente a la creación de la firma digital.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Notas de uso	Puede incluir la fecha y hora en que se ha generado la firma digital, el número de serie del hardware criptográfico utilizado o cualquier otra información relacionada con la creación de la firma. Los repositorios probablemente optarán por definir una estructura detallada apropiada para el contenido de <i>signatureProperties</i> .		

Unidad semántica	1.9.1.7 keyInformation		
Componentes semánticos	Contenedor de extensión		
Definición	Información referente a la clave pública del firmante, necesaria para validar la firma digital.		
Justificación	Para verificar la Firma Digital primero se recalcula el mensaje cifrado para el objeto y después se utiliza la clave pública del firmante para verificar que el valor de la firma (<i>signatureValue</i>) es correcto. Por lo tanto el repositorio debe tener el valor de la clave pública y asegurarse de algún modo de que realmente pertenece al firmante.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	No Aplicable	Aplicable	Aplicable
Repetible		No Repetible	No Repetible
Obligatoriedad		Opcional	Opcional
Notas de uso	Los diferentes tipos de claves tendrán diferentes estructuras y parámetros. PREMIS no define una estructura para este contenedor. Se recomienda representar los valores de las claves definidas para "KeyInfo" en W3C's <i>XML-Signature Syntax and Processing</i> (www.w3.org/TR/2002/REC-xmlsig-core-20020212/).		

Unidad semántica	1.9.2 signatureInformationExtension		
Componentes semánticos	Definido externamente		
Definición	Información de la firma digital procedente de unidades semánticas definidas fuera del diccionario PREMIS.		
Justificación	En ocasiones es necesario remplazar o ampliar las unidades semánticas definidas en el diccionario de datos PREMIS.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad		Aplicable	Aplicable
Repetible		Repetible	Repetible
Obligatoriedad		Opcional	Opcional
Notas de uso	<p>La extensibilidad se proporciona para un mayor nivel de detalle o para el uso de unidades semánticas definidas fuera del Diccionario. Se pueden incluir tanto unidades semánticas locales como metadatos procedentes de otro esquema de metadatos bien en adición bien en sustitución de las unidades semánticas definidas en el Diccionario PREMIS. Cuando se utilice un esquema de extensión, se debe proporcionar una referenciar al esquema utilizado. Para obtener más información relativa a la "extensibilidad" véase Capacidad de extensión.</p> <p>Si el contenedor <i>signatureInformationExtension</i> tiene que estar necesariamente asociado con alguna de las subunidades PREMIS de <i>signatureInformation</i>, se repite el contenedor <i>signatureInformation</i>. Si es necesario recurrir a la <i>extensibilidad</i> mediante esquemas de metadatos externos, el contenedor <i>signatureInformation</i> también debe repetirse. Se recomienda el uso de W3C's <i>XML-Signature Syntax and Processing</i> (www.w3.org/TR/2002/REC-xmldsig-core-20020212/) siempre que se pueda aplicar.</p>		

Unidad semántica	1.10 relationship		
Componentes semánticos	1.10.1 relationshipType 1.10.2 relationshipSubType 1.10.3 relatedObjectIdentification 1.10.4 relatedEventIdentification		
Definición	Información de la relación existente entre este objeto digital y otro u otros objetos digitales diferentes.		
Justificación	En un repositorio de preservación digital es necesario conocer como vincular los objetos complejos con cada una de sus partes (relaciones estructurales) así como registrar las relaciones de procedencia (relaciones de derivados). La documentación acerca de las relaciones entre objetos diferentes es crucial.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas de uso	<p>La mayoría de los repositorios quieren registrar todas las relaciones relevantes.</p> <p>En escenarios muy complejos, los metadatos PREMIS no aportan información suficiente para establecer con claridad las relaciones estructurales. Para ello existe gran variedad de metadatos estructurales, que proporcionan esta información de una manera más óptima que las unidades semánticas de los PREMIS.</p> <p>Las relaciones estructurales a nivel de fichero son necesarias para reconstruir las Representaciones y asegurar que dicha Representación es fiable.</p> <p>Por otra parte las relaciones estructurales a escala de cadena de bits permiten relacionar los bits dentro de un fichero en su correcta secuencia.</p> <p>Las relaciones de derivados a nivel de fichero y de representación son importantes para documentar la procedencia digital.</p>		

Unidad semántica	1.10.1 relationshipType		
Componentes semánticos	Ninguno		
Definición	Clasificación de la naturaleza de la relación.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicable	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Notas de uso	<p>Valores sugeridos: estructural = relación entre partes de un objeto derivation = relación en la que un objeto es el resultado de una transformación sobre otro objeto relacionado Los repositorios pueden considerar la utilidad de definir tipos adicionales.</p>		

Unidad semántica	1.10.2 relationshipSubType		
Componentes semánticos	Ninguno		
Definición	Categorización específica de la naturaleza de la relación documentada en el metadato anterior.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	<p>Algunos de los valores que se pueden utilizar son:</p> <p>has sibling = Este objeto comparte un <i>padre</i> común con el objeto relacionado</p> <p>is part of = Este objeto forma parte y está contenido dentro del objeto relacionado (cuando ambos son el mismo tipo de entidad)</p> <p>has part = Este objeto contiene al objeto relacionado (cuando ambos son el mismo tipo de entidad)</p> <p>is source of = El objeto relacionado es una versión de este objeto, que se ha creado como resultado de una transformación</p> <p>has source = Este objeto tiene una versión, que se ha creado como resultado de una transformación</p> <p>has root = Sólo se utiliza en las REPRESENTACIONES; Significa que el objeto relacionado es el fichero o CADENA DE BITS que debe ser procesado en primer lugar para que la representación final del objeto sea correcta</p> <p>includes = Se utiliza para indicar la relación entre una Representación y un Archivo o bien entre un Archivo y un CADENA DE BITS. El objeto descrito incluye al objeto referenciado</p> <p>is included in = Se utiliza para indicar la relación entre un Archivo y una Representación o bien entre un CADENA DE BITS y un Archivo. El objeto descrito está incluido en el objeto referenciado</p> <p>El repositorio puede considerar necesario definir relaciones más o menos detalladas. Para las relaciones de derivación hay que destacar que la relación precisa puede estar indicada por el tipo de acontecimiento relacionado.</p> <p>La relación "has root" se aplica sólo a representaciones porque implica que un objeto compuesto (por ejemplo: uno formado por ficheros múltiples) requiere que se escoja primero un fichero como su raíz. En los metadatos de la representación "has root" identifica ese fichero en concreto.</p>		

Unidad semántica	1.10.3 relatedObjectIdentification		
Componentes semánticos	1.10.3.1 relatedObjectIdentifierType 1.10.3.2 relatedObjectIdentifierValue 1.10.3.3 relatedObjectSequence		
Definición	Identificador y contexto de los objetos relacionados.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Notas de uso	Los objetos relacionados pueden o no estar albergados en el repositorio digital. La práctica recomendada es que el objeto resida en el repositorio a no ser que exista una buena razón para lo contrario. En ambos casos la referencia al objeto relacionado debe ser muy clara.		

Unidad semántica	1.10.3.1 relatedObjectIdentifierType		
Componentes semánticos	Ninguna		
Definición	Designación del dominio en el que el identificador es único.		
Limitaciones	El término debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	DLC DRS [ejemplos de <i>ObjectIdentifierType</i>]	DLC DRS [ejemplos de <i>ObjectIdentifierType</i>]	DLC DRS [ejemplos de <i>ObjectIdentifierType</i>]
Notas de uso	Si el objeto relacionado se conserva dentro del Repositorio digital, este debería ser el valor <i>ObjectIdentifierType</i> de ese objeto.		

Unidad semántica	1.10.3.2 relatedObjectIdentifierValue		
Componentes semánticos	Ninguno		
Definición	Valor del Identificador del objeto digital relacionado.		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No repetible	No repetible	No repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	[Ver ejemplos de: <i>objectIdentifierValue</i>]	[Ver ejemplos de: <i>objectIdentifierValue</i>]	[Ver ejemplos de: <i>objectIdentifierValue</i>]
Notas de uso	Si el objeto relacionado se conserva dentro del Repositorio digital, este debería ser el valor <i>ObjectIdentifierValue</i> de ese objeto.		

Unidad semántica	1.10.3.3 relatedObjectSequence		
Componentes semánticos	Ninguno		
Definición	El orden del objeto relacionado con respecto a otros objetos con los que comparte relación.		
Justificación	Esta unidad semántica es particularmente útil en las relaciones estructurales. Para reconstruir una representación, es necesario conocer el orden de los componentes en relaciones de hermanos o de todo/parte. Por ejemplo, para reconstruir un libro cuyas páginas sean imágenes es necesario conocer el orden de los ficheros que representan las páginas.		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo		1 2 3	
Notas de uso	<p>Esta unidad semántica puede aplicarse de diferentes maneras. Puede registrarse en los metadatos como número secuencial o como un puntero. Puede estar implícita en algún otro orden de objetos, por ejemplo, incrementando valores de los identificadores. El valor de <i>relationshipSubType</i> podría implicar la secuencia (por ejemplo: "is preceding sibling", "is following sibling").</p> <p>No es obligatorio que los números sean únicos o secuenciales. Algunos objetos relacionados no cuentan con orden secuencial, por ejemplo las páginas Web que forman un sitio Web. En este caso todos los objetos relacionados pueden tener el número secuencial cero.</p> <p>Esta unidad semántica se aplica solo a relaciones estructurales y por lo tanto es opcional.</p>		

Unidad semántica	1.10.4 relatedEventIdentification		
Componentes semánticos	1.10.4.1 relatedEventIdentifierType 1.10.4.2 relatedEventIdentifierValue 1.10.4.3 relatedEventSequence		
Definición	Identificador y secuencia contextual de un Acontecimiento asociado a una relación entre dos o más objetos.		
Justificación	La relación entre dos objetos puede venir determinada por un acontecimiento como es por ejemplo una Migración.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo		1 2 3	
Notas de uso	<i>relatedEventIdentification</i> debe ser cumplimentado siempre que existan relaciones derivadas entre objetos.		

Unidad semántica	1.10.4.1 relatedEventIdentifierType		
Componentes semánticos	Ninguno		
Definición	El EventIdentifierType del evento relacionado.		
Justificación	Debe existir a priori un valor EventIdentifierType.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	[véanse los ejemplos para: <i>eventIdentifierType</i>]	[véanse los ejemplos para: <i>eventIdentifierType</i>]	[véanse los ejemplos para: <i>eventIdentifierType</i>]
Notas de uso	En la mayoría de repositorios de preservación, <i>eventIdentifierType</i> es simplemente un sistema de numeración propio. Puede ser un sistema implícito que sólo aparece explícitamente si exportamos los datos.		

Unidad semántica	1.10.4.2 relatedEventIdentifierValue		
Componentes semánticos	Ninguno		
Definición	El <i>eventIdentifierValue</i> del evento relacionado.		
Justificación	Debe existir a priori un <i>eventIdentifierValue</i> .		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplos	[Véanse los ejemplos para: <i>eventIdentifierValue</i>]	[Véanse los ejemplos para: <i>eventIdentifierValue</i>]	[Véanse los ejemplos para: <i>eventIdentifierValue</i>]

Unidad semántica	1.10.4.3 relatedEventSequence		
Componentes semánticos	Ninguno		
Definición	Orden del evento relacionado		
Limitaciones	Ninguna		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Ejemplo		1 2 3	
Notas de uso	El orden de secuencias de un evento relacionado se puede obtener del valor <i>eventDateTime</i> asociado al evento relacionado.		

Unidad semántica	1.11 linkingEventIdentifier		
Componentes semánticos	1.11.1 linkingEventIdentifierType 1.11.2 linkingEventIdentifierValue		
Definición	El <i>EventIdentifier</i> de un evento asociado a un objeto.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas de uso	Se utiliza para enlazar acontecimientos que no están asociados con relaciones entre objetos como validaciones de formatos, comprobación de virus, etc.		

Unidad semántica	1.11.1 linkingEventIdentifierType		
Componentes semánticos	Ninguno		
Definición	El <i>EventIdentifierType</i> de un evento asociado a un objeto.		
Limitaciones	Debe existir a priori una tipología de valores de <i>eventIdentifierType</i> .		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	[Véanse los ejemplos para <i>eventIdentifierType</i>]	[Véanse los ejemplos para <i>eventIdentifierType</i>]	[Véanse los ejemplos para <i>eventIdentifierType</i>]
Notas de uso	Se utiliza para enlazar acontecimientos que no están asociados con relaciones entre objetos como validaciones de formatos, comprobación de virus, etc.		

Unidad semántica	1.11.2 linkingEventIdentifierValue		
Componentes semánticos	Ninguno		
Definición	El <i>eventIdentifierValue</i> del evento relacionado.		
Justificación	Debe existir a priori una tipología de valores de <i>eventIdentifierValue</i> .		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	[Véanse los ejemplos para <i>eventIdentifierValue</i>]	[Véanse los ejemplos para <i>eventIdentifierValue</i>]	[Véanse los ejemplos para <i>eventIdentifierValue</i>]

Unidad semántica	1.12 linkingIntellectualEntityIdentifier		
Componentes semánticos	1.12.1 linkingIntellectualEntityIdentifierType 1.12.2 linkingIntellectualEntityIdentifierValue		
Definición	Identificador para la entidad intelectual asociada con el objeto.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional
Notas de uso	Se utiliza para vincular el objeto con una entidad intelectual que esté relacionada con el mismo. Por ejemplo puede ser un vínculo o link a los metadatos descriptivos que describen esta entidad intelectual relacionada con el objeto. Este vínculo será probablemente el identificador de un objeto que se encuentre en un nivel conceptual mayor que el objeto por ejemplo, a una colección de objetos.		

Unidad semántica	1.12.1 linkingIntellectualEntityIdentifierType		
Componentes semánticos	Ninguno		
Definición	Designación del dominio en el que el <i>linkingIntellectualEntityIdentifier</i> es único.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo		URI LCCN	

Unidad semántica	1.12.2 linkingIntellectualEntityIdentifierValue		
Componentes semánticos	Ninguno		
Definición	Valor del <i>linkingIntellectualEntityIdentifier</i> .		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo	hdl:loc.natlib/mrva00 02.0495 info:lccn/19018302		

Unidad semántica	1.13 linkingRightsStatementIdentifier		
Componentes semánticos	1.13.1 linkingRightsStatementIdentifierType 1.13.2 linkingRightsStatementIdentifierValue		
Definición	Identificador para la declaración de derechos asociados al objeto.		
Justificación	El repositorio decide si enlaza desde una declaración de derechos a un objeto o desde el objeto a la declaración o ambos casos.		
Limitaciones	Contenedor		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	Repetible	Repetible	Repetible
Obligatoriedad	Opcional	Opcional	Opcional

Unidad semántica	1.13.1 linkingRightsStatementIdentifierType		
Componentes semánticos	Ninguno		
Definición	Designación del dominio en el que el <i>linkingRightsStatementIdentifier</i> es único.		
Limitaciones	El valor debe proceder de un vocabulario controlado.		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio
Ejemplo		URI LCCN	

Unidad semántica	1.13.2 linkingRightsStatementIdentifierValue		
Componentes semánticos	Ninguno		
Definición	El valor del <i>linkingRightsStatementIdentifier</i> .		
Limitaciones	Ninguno		
Categoría	Representación	Fichero	Cadena de bits
Aplicabilidad	Aplicable	Aplicable	Aplicable
Repetible	No Repetible	No Repetible	No Repetible
Obligatoriedad	Obligatorio	Obligatorio	Obligatorio

2.3 Entidad Acontecimiento

La entidad Acontecimiento contiene información sobre una acción que afecta a una o a varias entidades Objeto.

Los repositorios registran información sobre Acontecimientos dependiendo de la importancia del acontecimiento. Las Acciones que modifican objetos deberían registrarse siempre, otras acciones como las copias de seguridad pueden quedar registradas en listados del sistema o similares pero no en una entidad Acontecimiento.

Las unidades semánticas obligatorias son: *eventIdentifier*, *eventType*, y *eventDateTime*.

2.3.1 Propiedades de la Entidad

- Debe afectar a uno o a varios objetos.
- Puede estar relacionada con uno o con varios agentes.
- Los enlaces entre entidades pueden registrarse desde cualquier dirección y no es necesario que sean bidireccionales.

2.3.2 Unidades semánticas de la Entidad

2.1 eventIdentifier (identificador del acontecimiento) (O, NR)

2.1.1 eventIdentifierType (tipo de identificador del acontecimiento) (O, NR)

2.1.2 eventIdentifierValue (valor del identificador del acontecimiento) (O, NR)

2.2 eventType (tipo de acontecimiento) (O, NR)

2.3 eventDateTime (tiempo de la fecha del acontecimiento) (O, NR)

2.4 eventDetail (detalle del acontecimiento) (M, NR)

2.5 eventOutcomeInformation (información sobre las consecuencias del acontecimiento) (M, R)

2.5.1 eventOutcome (consecuencias del acontecimiento) (M, NR)

2.5.2 eventOutcomeDetail (detalles sobre las consecuencias del acontecimiento) (M, R)

2.5.2.1 eventOutcomeDetailNote (nota sobre los detalles sobre las consecuencias del acontecimiento) (M, NR)

2.5.2.2 eventOutcomeDetailExtension (Extensión de los detalles sobre las consecuencias del acontecimiento) (M, R)

2.6 linkingAgentIdentifier (identificador del agente vinculado) (M, R)

2.6.1 linkingAgentIdentifierType (tipo de identificador del agente vinculado) (O, NR)

2.6.2 linkingAgentIdentifierValue (valor del identificador del agente vinculado) (O, NR)

2.6.3 linkingAgentRole (función del agente vinculado) (M, R)

2.7 linkingObjectIdentifier (identificador del objeto vinculado) (M, R)

2.7.1 linkingObjectIdentifierType (tipo de identificador del objeto vinculado) (O, NR)

2.7.2 linkingObjectIdentifierValue (valor del identificador del objeto vinculado) (O, NR)

2.7.3 linkingObjectRole (función del objeto vinculado) (M, R)

Unidad semántica	2.1 eventIdentifier
Componentes semánticos	2.1.1 eventIdentifierType 2.1.2 eventIdentifierValue
Definición	Denominación para identificar unívocamente el acontecimiento en el seno del repositorio.
Justificación	Cada acontecimiento registrado en el repositorio debe tener un identificador único que permita asociarlo de manera inequívoca a objetos, agentes y otros acontecimientos.
Limitaciones	Contenedor
Repetible	No Repetible
Obligatoriedad	Obligatorio
Notas de creación/mantenimiento	Es probable que el sistema del repositorio genere automáticamente <i>eventIdentifier</i> . No hay una norma internacional acordada para estos identificadores de manera que el identificador no es repetible.

Unidad semántica	2.1.1 eventIdentifierType
Componentes semánticos	Ninguno
Definición	Nombre del dominio en el cual el identificador del acontecimiento es unívoco.
Limitación	Ninguna
Repetible	No Repetible
Obligatoriedad	Obligatorio
Ejemplos	FDA Stanford Repository Event ID UUID: Universally Unique Identifier
Notas mantenimiento	Es probable que el sistema del repositorio genere automáticamente <i>eventIdentifierType</i> . No hay una norma internacional acordada para estos identificadores de manera que el identificador no es repetible.

Unidad semántica	2.1.2 eventIdentifierValue
Componentes semánticos	Ninguno
Definición	El valor de <i>eventIdentifier</i> .
Limitación	Ninguna
Repetible	No Repetible
Obligatoriedad	Obligatorio
Ejemplos	E-2004-11-13-000119 58f202ac-22cf-11d1-b12d-002035b29092

Unidad semántica	2.2 eventType
Componentes semánticos	Ninguno
Definición	Categorización de la naturaleza de un acontecimiento.
Justificación	La categorización de los acontecimientos ayudará a los repositorios en el procesado automático de la información sobre el acontecimiento, sobre todo para la elaboración de informes.
Limitaciones	El valor debería proceder de un vocabulario controlado.
Repetible	No Repetible
Obligatoriedad	Obligatorio
Ejemplo	E77 [código utilizado dentro de un repositorio para identificar un Acontecimiento en concreto] Ingest
Notas de creación/mantenimiento	<p>Cada Repositorio debería definir su propio vocabulario controlado para la categorización de los valores de <i>eventType</i>. No obstante, a continuación se referencia un posible listado inicial de valores (Véase también el glosario para más detalles):</p> <p>Captura= proceso mediante el cual un repositorio obtiene un objeto de forma activa.</p> <p>Compresión= proceso de codificación de datos para ahorrar espacio de almacenamiento o tiempo de transmisión de datos</p> <p>Creación = acto de crear un nuevo objeto</p> <p>Eliminación= proceso de eliminación de un objeto del inventario del repositorio</p> <p>Descompresión= proceso de revertir los efectos de la compresión</p> <p>Descifre= proceso de conversión a texto de los datos cifrados</p> <p>Borrado= proceso de eliminación de un objeto del depósito de almacenamiento</p> <p>Validación de firma digital= proceso mediante el cual se determina que tras decodificar una firma digital el valor obtenido corresponde con el valor esperado</p> <p>Difusión= proceso de recuperación de un objeto del repositorio de almacenamiento para ponerlo a disposición de los usuarios</p> <p>Fixity check= proceso por el que se verifica que un objeto no ha sido modificado en un período determinado</p>

	<p>Ingest= proceso de agregar objetos a un repositorio de preservación</p> <p>message digest calculation = proceso por el que se crea un mensaje cifrado ("hash")</p> <p>Migración= transformación de un objeto a una nueva versión cuyo formato sea más actual</p> <p>Normalización= transformación de un objeto creando una versión más adecuada para la preservación</p> <p>Replicar Duplicar= Proceso de crear una copia idéntica al original</p> <p>Validación= proceso de comparar un objeto con un estándar para analizar si existen diferencias</p> <p>Análisis de virus= proceso de escaneado de un objeto en busca de virus ocultos</p> <p>Observación: migración, normalización y réplica son subtipos más precisos del acontecimiento creación. "Creación" puede utilizarse cuando no haya términos más precisos, por ejemplo, cuando un objeto digital se haya creado escaneando el documento de papel.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Unidad semántica	2.3 eventDateTime
Componentes semánticos	Ninguno
Definición	La fecha y la hora exacta o bien rango de las mismas durante el cual se ha producido un acontecimiento.
Limitaciones	Este valor debería adoptar una forma estructurada para facilitar el proceso automatizado de los datos. Se recomienda utilizar convenciones normalizadas que faciliten el intercambio de metadatos conforme a PREMIS, por ejemplo las utilizadas en el esquema PREMIS.
Repetible	No Repetible
Obligatoriedad	Obligatorio
Ejemplos	20050704T071530-0500 [July 4, 2005 at 7:15:30 a.m. EST] 2006-07-16T19:20:30+01:00 20050705T0715-0500/20050705T0720-0500 [from 7:15 a.m. EST to 7:20 a.m. EST on July 4, 2005] 2004-03-17 [March 17, 2004, solo se conoce la fecha]
Notas mantenimiento	Se recomienda poner la fecha y hora de la manera más exacta posible así como incluir la zona horaria que se toma como referencia.

Unidad semántica	2.4 eventDetail
Componentes semánticos	Ninguno
Definición	Información adicional del acontecimiento.
Limitaciones	Ninguno
Repetible	No Repetible
Obligatoriedad	Opcional
Ejemplos	Objeto retirado permanentemente a petición del autor.
Notas de uso	La información recogida en <i>eventDetail</i> no está destinada a ser procesados por la máquina. De tal manera que se puede registrar cualquier tipo de información referente a un acontecimiento y/o referirse a información almacenada en otro lugar.

Unidad semántica	2.5 eventOutcomeInformation
Componentes semánticos	2.5.1 eventOutcome 2.5.2 eventOutcomeDetail
Definición	Información referente al resultado de un acontecimiento.
Limitaciones	Contenedor
Repetible	Repetible
Obligatoriedad	Opcional
Nota de uso	Un repositorio podría desear añadir información al valor codificado <i>eventOutcome</i> en <i>eventOutcomeDetail</i> . El contenedor es repetible ya que los acontecimientos pueden producir más de un resultado. Todas las subunidades de esta unidad semántica son opcionales pero por lo menos una de las subunidades (<i>eventOutcome</i> o <i>eventOutcomeDetail</i>) debe estar presente si se incluye este contenedor.

Unidad semántica	2.5.1 eventOutcome
Componentes semánticos	Ninguno
Definición	Una categorización del resultado global del acontecimiento en términos de éxito, éxito parcial, o fracaso.
Justificación	Un código que permita representar el resultado de un acontecimiento es muy útil para el procesamiento automático y para realizar informes del proceso. Por ejemplo, si un control de fiabilidad fracasa, el registro de acontecimiento proporciona un registro con información permanente pero que puede modificarse.
Limitaciones	El valor debe proceder de un vocabulario controlado.
Repetible	No Repetible
Obligatoriedad	Opcional
Ejemplos	00 [código cuyo significado es "acción completada con éxito"] CV-01 [código cuyo significado es "verificación de datos validada"]
Notas mantenimiento	Se recomienda utilizar un vocabulario controlado, de manera que el sistema actúe automáticamente. Se puede utilizar el elemento <i>eventOutcomeDetail</i> para registrar más detalles acerca del resultado del acontecimiento. Se recomienda definir los acontecimientos con el suficiente nivel de detalle para que cada acontecimiento tenga un solo resultado.

Unidad semántica	2.5.2 eventOutcomeDetail
Componentes semánticos	2.5.2.1 eventOutcomeDetailNote 2.5.2.2 eventOutcomeDetailExtension
Definición	Descripción detallada del resultado del acontecimiento.
Justificación	El resultado de un acontecimiento puede ser muy complejo, de forma que un código de descripción puede no ser suficiente para documentar el resultado del proceso.
Limitaciones	Contenedor
Repetible	Repetible
Obligatoriedad	Opcional
Notas de uso	<p>Se puede utilizar para registrar todos los mensajes de error y advertencia emitidos por el programa que gestiona el acontecimiento o bien para registrar un enlace a un registro de errores.</p> <p>Si el acontecimiento es un análisis de validación cualquier anomalía detectada se registraría en esta unidad.</p> <p>Todas las subunidades de esta unidad semántica son opcionales pero por lo menos una de las subunidades (<i>eventOutcomeDetailNote</i> y/o <i>eventOutcomeDetailExtension</i>) debe estar presente si se incluye este contenedor.</p>

Unidad semántica	2.5.2.1 eventOutcomeDetailNote
Componentes semánticos	Ninguno
Definición	Descripción detallada y textual del resultado del acontecimiento.
Justificación	En ocasiones es importante incorporar información textual adicional referente al resultado del acontecimiento.
Limitaciones	Ninguna
Repetible	No Repetible
Obligatoriedad	Opcional
Ejemplo	LZW archivo comprimido No se han encontrado etiquetas normalizadas en la cabecera

Unidad semántica	2.5.2.2 eventOutcomeDetailExtension
Componentes semánticos	Definido externamente
Definición	Un Contenedor que incluye unidades semánticas definidas fuera del Diccionario de Datos PREMIS.
Justificación	Puede ser necesario remplazar o completar las unidades definidas en el Diccionario de Datos PREMIS.
Limitaciones	Contenedor
Repetible	Repetible
Obligatoriedad	Opcional
Notas de uso	<p>La extensibilidad se proporciona para un mayor nivel de detalle o para el uso de unidades semánticas definidas fuera del Diccionario de Datos. Se pueden incluir tanto unidades semánticas locales como metadatos procedentes de otro esquema de metadatos bien en adición bien en sustitución de las unidades semánticas definidas en el Diccionario PREMIS.</p> <p>Cuando se utilice un esquema de extensión, se debe proporcionar una referencia al esquema utilizado. Para obtener más información relativa a la "extensibilidad" véase capacidad de extensión.</p> <p>Si el contenedor <i>acontecimientooutcomeDetailExtension</i> tiene que estar necesariamente asociado con alguna de las subunidades PREMIS dependientes de <i>eventOutcomeDetail</i>, se repite el contenedor. Si es necesario recurrir a la <i>extensibilidad</i> mediante esquemas de metadatos externos, el contenedor <i>eventOutcomeDetail</i> también debe repetirse.</p>

Unidad semántica	2.6 linkingAgentIdentifier
Componentes semánticos	2.6.1 linkingAgentIdentifierType 2.6.2 linkingAgentIdentifierValue 2.6.3 linkingAgentRole
Definición	Información referente a un agente asociado a un acontecimiento.
Justificación	El concepto de "procedencia digital" a menudo requiere que las relaciones entre Agente y Acontecimiento estén documentadas.
Limitaciones	Contenedor
Repetible	Repetible
Obligatoriedad	Opcional
Notas de uso	Se recomienda, si es posible, registrar al Agente.

Unidad semántica	2.6.1 linkingAgentIdentifierType
Componentes semánticos	Ninguno
Definición	Nombre del dominio en el cual el identificador del agente asociado es único.
Limitaciones	El valor debe proceder de un vocabulario controlado.
Ejemplos	[Véanse los ejemplos para agentIdentifierType]
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	2.6.2 linkingAgentIdentifierValue
Componentes semánticos	Ninguno
Definición	Valor del identificador del agente vinculado.
Limitaciones	Ninguno
Ejemplos	[Véanse los ejemplos para agentIdentifierValue]
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	2.6.3 linkingAgentRole
Componentes semánticos	Ninguno
Definición	La función que desempeña el agente en este acontecimiento.
Justificación	Los acontecimientos pueden tener más de un agente asociado. El papel de cada agente debería ser documentado.
Limitaciones	Los valores deben proceder de un vocabulario controlado.
Repetible	Repetible
Obligatoriedad	Opcional
Ejemplos	Authorizer Implementer Validator Executing program

Unidad semántica	2.7 linkingObjectIdentifier
Componentes semánticos	2.7.1 linkingObjectIdentifierType 2.7.2 linkingObjectIdentifierValue 2.7.3 linkingObjectRole
Definición	Información acerca de un objeto asociado con un Acontecimiento.
Justificación	La "Digital Provenance" (procedencia digital o historia del objeto) a menudo requiere que las relaciones ente objetos y Acontecimientos están documentadas.
Limitaciones	Contenedor
Repetible	Repetible
Obligatoriedad	Opcional

Unidad semántica	2.7.1 linkingObjectIdentifierType
Componentes semánticos	Ninguno
Definición	Nombre del dominio en el cual el identificador del objeto asociado es único.
Limitaciones	El valor debe proceder de un vocabulario controlado.
Ejemplos	[véanse los ejemplos para objectIdentifierType]
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	2.7.2 linkingObjectIdentifierValue
Componentes semánticos	Ninguno
Definición	Valor del identificador del objeto vinculado.
Limitaciones	Ninguno
Ejemplos	[véanse los ejemplos para objectIdentifierValue]
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	2.7.3 linkingObjectRole
Componentes semánticos	Ninguno
Definición	El rol o función que desempeña el objeto asociado a un acontecimiento.
Justificación	Distingue la función del objeto en relación a un acontecimiento. Si la función del objeto no está explícita, entonces será necesario analizar la relación entre los objetos en los propios metadatos del objeto.
Limitaciones	Ninguno
Repetible	Repetible
Obligatoriedad	Opcional
Ejemplo	Source = origen Outcome = resultado

2.4 Entidad Agentes

La entidad Agentes proporciona información acerca de los atributos o características de los agentes (personas, organizaciones, o software) vinculados a los derechos de gestión y a los acontecimientos relativos a la preservación y al ciclo de vida de un objeto. La información vinculada al Agente lo identifica unívocamente del resto de entidades.

La única unidad semántica obligatoria es *agentIdentifier*.

2.4.1 Propiedades de la Entidad

- Puede sustentar o conceder uno o más derechos.
- Puede realizar, autorizar, o exigir a uno o más acontecimientos.
- Puede crear o actuar sobre uno o más objetos a través de un acontecimiento o con respecto a una declaración de derechos

2.4.2 Unidades semánticas de la Entidad

3.1 *agentIdentifier* (identificador del agente) (O, R)

3.1.1 *agentIdentifierType* (tipo de identificador del agente)

3.1.2 *agentIdentifierValue* (valor del identificador del agente)

3.2 *agentName* (nombre del agente) (M, R)

3.3 *agentType* (tipo de agente) (M, NR)

Unidad semántica	3.1.agentIdentifier
Componentes semánticos	3.1.1 agentIdentifierType 3.1.2 agentIdentifierValue
Definición	Denominación que se utiliza para identificar unívocamente el <i>agente</i> en el repositorio de preservación.
Justificación	Cada Agente registrado en el repositorio debe tener un identificador único que permita su asociación, de manera inequívoca, a acontecimientos y declaraciones de derechos.
Limitaciones	Ninguna
Repetible	No Repetible
Obligatoriedad	Obligatorio
Notas de creación/mantenimiento	Es posible que el sistema del repositorio genere los identificadores automáticamente, o bien que éstos puedan ser generados de manera externa. Por otra parte, los identificadores pueden ser generados automáticamente o de forma manual. No hay una norma internacional acordada para esta actividad. No obstante, la práctica recomienda que lo más seguro es que los identificadores se generen automáticamente y por el mismo repositorio. No se descarta la posibilidad de asignar identificadores asignados externamente de forma que se puedan utilizar de manera secundaria para vincular al Agente con información referente al mismo localizada fuera del repositorio.
Notas de uso	Los identificadores deben ser únicos dentro del repositorio. Como se puede observar, esta unidad semántica es repetible ya que pueden utilizarse identificadores externos.

Unidad semántica	3.1.1 agentIdentifierType
Componentes semánticos	Ninguno
Definición	Nombre del dominio en el cual el identificador del agente es unívoco.
Limitaciones	El valor debe proceder de un listado controlado de términos.
Ejemplos	LCNAF SAN MARC Organization Codes URI
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	3.1.2 agentIdentifierValue
Componentes semánticos	Ninguno
Definición	El valor del <i>agentIdentifier</i> .
Limitaciones	Ninguna
Repetible	No Repetible
Obligatoriedad	Obligatorio
Ejemplos	92-79971 Owens, Erik C. 234-5676 MH-CS Info:lccn/n78890351
Notas de uso	Puede ser un código o bien un nombre controlado conforme a una lista de autoridades.

Unidad semántica	3.2 agentName
Componentes semánticos	Ninguno
Definición	Cadena de texto que se puede utilizar para añadir información a la unidad semántica del <i>agentIdentifier</i> con el fin de identificar unívocamente al agente.
Justificación	Esta unidad semántica proporciona mayor interpretación lectora del identificador del agente que la que proporciona el <i>agentIdentifier</i> .
Limitaciones	Ninguna
Repetible	Repetible
Ejemplos	Erik Owens Woodyard Pc
Obligatoriedad	Opcional
Notas de uso	El valor no tiene por qué ser necesariamente único.

Unidad semántica	3.3 agentType
Componentes semánticos	Ninguno
Definición	Caracterización concisa del tipo de agente.
Limitaciones	El valor debe proceder de un listado controlado de términos.
Repetible	No repetible
Obligatoriedad	Opcional
Notas de uso	Posibles valores: persona organización software

2.5 Entidad Derechos

A efectos del Diccionario de Datos PREMIS, se considera que tanto las declaraciones de derechos como los permisos son construcciones que se pueden describir en la entidad de Derechos. Los Derechos se confieren por los agentes mediante derechos de autor u otra tipología de derecho de propiedad intelectual. Los permisos son poderes o privilegios otorgados por acuerdo entre un titular de derechos y una o más partes.

En un repositorio puede quedar registrada una gran variedad de información sobre los derechos, desde resúmenes de declaraciones de derecho hasta los permisos y derechos de agentes externos y objetos no depositados en el repositorio. Sin embargo, el corpus mínimo de información de derechos que un repositorio de preservación debería conocer, es el que hace referencia a los derechos o permisos con los que un repositorio tiene que cumplir y que corresponde a las acciones que afectan a los objetos dentro del repositorio. Esta información puede quedar garantizada por la ley de copyright, por una norma, o por un acuerdo de licencia con el titular de los derechos.

Si el repositorio conserva información relativa a los derechos, debe cumplimentarse bien el *rightsStatement* o bien el *rightsExtension*.

2.5.1 Propiedades de la Entidad

- Puede estar relacionada con uno o más objetos.
- Puede estar relacionada con uno o más agentes.
- Los enlaces entre las entidades pueden quedar registrados desde una única dirección y sin necesidad de ser bidireccionales.

2.5.2 Unidades Semánticas de la Entidad

4.1 rightsStatement (mención de derechos) (M, R)

4.1.1 rightsStatementIdentifier (identificador de la mención de derechos) (O, NR)

4.1.1.1 rightsStatementIdentifierType (tipo de identificador de la mención de derechos) (O, NR)

4.1.1.2 rightsStatementIdentifierValue (valor del identificador de la mención de derechos) (O, NR)

4.1.2 rightsBasis (bases de los derechos) (O, NR)

4.1.3 copyrightInformation (información sobre el copyright) (M, NR)

4.1.3.1 copyrightStatus (estado del copyright) (O, NR)

4.1.3.2 copyrightJurisdiction (jurisdicción del copyright) (O, NR)

4.1.3.3 copyrightStatusDeterminationDate (determinación de la fecha del estado del copyright) (M, NR)

4.1.3.4 copyrightNote (nota sobre el copyright) (M, R)

4.1.4 licenseInformation (información sobre la licencia) (M, NR)

4.1.4.1 licenseIdentifier (identificador de la licencia) (M, NR)

- 4.1.4.1.1 licenseIdentifierType (tipo del identificador de la licencia) (O, NR)
- 4.1.4.1.2 licenseIdentifierValue (valor del identificador de la licencia) (O, NR)
- 4.1.4.2 licenseTerms (términos de la licencia) (M, NR)
- 4.1.4.3 licenseNote (nota sobre la licencia) (M, R)
- 4.1.5 statuteInformation (información sobre la legislación) (M, R)
 - 4.1.5.1 statuteJurisdiction (jurisdicción de la legislación) (O, NR)
 - 4.1.5.2 statuteCitation (cita de la legislación) (O, NR)
 - 4.1.5.3 statuteInformationDeterminationDate (fecha de determinación de la información sobre la legislación) (M, NR)
 - 4.1.5.4 statuteNote (nota sobre la legislación) (M, R)
- 4.1.6 rightsGranted (derechos otorgados) (M, R)
 - 4.1.6.1 act (ley) (O, NR)
 - 4.1.6.2 restriction (restricción) (M, R)
 - 4.1.6.3 termOfGrant (período por el que se otorgan los derechos) (O, NR)
 - 4.1.6.3.1 startDate (fecha de inicio) (O, NR)
 - 4.1.6.3.2 endDate (fecha final) (M, NR)
 - 4.1.6.4 rightsGrantedNote (nota sobre los derechos otorgados) (M, R)
- 4.1.7 linkingObjectIdentifier (identificador del objeto vinculado) (M, R)
 - 4.1.7.1 linkingObjectIdentifierType (tipo de identificador del objeto vinculado) (O, NR)
 - 4.1.7.2 linkingObjectIdentifierValue (valor del identificador del objeto vinculado) (O, NR)
- 4.1.8 linkingAgentIdentifier (identificador del agente vinculado) (M, R)
 - 4.1.8.1 linkingAgentIdentifierType (tipo de identificador del agente vinculado) (O, NR)
 - 4.1.8.2 linkingAgentIdentifierValue (valor del identificador del agente vinculado) (O, NR)
 - 4.1.8.3 linkingAgentRole (función del agente vinculado) (O, NR)
- 4.2 rightsExtension (extensión de los derechos) (M, R)

Unidad semántica	4.1 rightsStatement
Componentes semánticos	<p>4.1.1 rightsStatementIdentifier 4.1.2 rightsBasis 4.1.3 copyrightInformation 4.1.4 licenseInformation 4.1.5 statuteInformation 4.1.6 rightsGranted 4.1.7 linkingObjectIdentifier 4.1.8 linkingAgentIdentifier</p>
Definición	Documentación que permite identificar los derechos que posee o han sido otorgados al repositorio para realizar ciertas actividades.
Limitaciones	Container
Repetible	Repetible
Obligatoriedad	Opcional
Notas de mantenimiento	<p>Esta unidad semántica es opcional ya que en algunos casos los derechos pueden ser desconocidos. No obstante, se debe motivar que las instituciones registren información sobre derechos siempre que sea posible.</p> <p>Siempre que se incluya la entidad Rights debe aparecer o bien <i>rightsStatement</i> o bien <i>rightsExtension</i>.</p> <p><i>rightsStatement</i> deberá repetirse en aquellos casos en los que la normativa describa más de un supuesto o bien cuando diferentes normas se sustenten sobre diferentes bases.</p>

Unidad semántica	4.1.1 rightsStatementIdentifier
Componentes semánticos	4.1.1.1 rightsStatementIdentifierType 4.1.1.2 rightsStatementIdentifierValue
Definición	Denominación que se utiliza para identificar unívocamente la declaración de derechos en el sistema del repositorio de preservación.
Justificación	Cada declaración de derechos registrada en el repositorio debe tener un identificador único que permita asociarlo de manera inequívoca a Acontecimientos y Agentes.
Limitaciones	Contenedor
Repetible	No Repetible
Obligatoriedad	Obligatorio
Notas de creación/mantenimiento	Es recomendable que el sistema del repositorio genere automáticamente el <i>rightsStatementIdentifier</i> . Sin embargo no existe un esquema global o normalizado para formalizar este tipo de identificadores. Este Identificador no es repetible.
Notas de uso	Los identificadores deben ser únicos dentro del repositorio.

Unidad semántica	4.1.1.1 rightsStatementIdentifierType
Componentes semánticos	Ninguno
Definición	Designación de dominio en el cual el identificador de la declaración de derechos es único.
Limitaciones	El valor debe proceder de un listado controlado de términos.
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.1.2 rightsStatementIdentifierValue
Componentes semánticos	Ninguno
Definición	Valor concreto del <i>rightsStatementIdentifier</i> .
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.2 rightsBasis
Componentes semánticos	Ninguno
Definición	Designación que sirve de base para los derechos o permisos descritos en el <i>rightsStatementIdentifier</i> .
Limitaciones	El valor debe proceder de un listado controlado de términos.
Repetible	No Repetible
Obligatoriedad	Obligatorio
Notas de mantenimiento	<p>Posibles valores: copyright, licencias, estatutos. Cuando <i>rightsBasis</i> es "copyright", siempre debe aparecer <i>copyrightInformation</i>. Cuando <i>rightsBasis</i> es "licencia", siempre debe aparecer <i>licenseInformation</i>. Cuando <i>rightsBasis</i> es "estatuto", siempre debe aparecer <i>statuteInformation</i>.</p> <p>Si la base de los derechos es el <i>dominio público</i>, debe utilizarse "copyright". Si la base es <i>uso legítimo</i>, utilícese "estatuto". Si se sustenta sobre más de una base, deberá repetirse la entidad de derechos en su totalidad.</p>

Unidad semántica	4.1.3 copyrightInformation
Componentes semánticos	4.1.3.1 copyrightStatus 4.1.3.2 copyrightJurisdiction 4.1.3.3 copyrightStatusDeterminationDate 4.1.3.4 copyrightNote
Definición	Información referente al status de los derechos de un objeto u objetos.
Limitaciones	Contenedor
Repetible	No Repetible
Obligatoriedad	Opcional
Notas de creación/mantenimiento	<p>Cuando el valor de <i>rightBasis</i> es <i>copyright</i>, <i>copyrightInformation</i> también debe aparecer cumplimentado. En ocasiones los repositorios necesitan ampliar o ser más concisos respecto a la información referente a los derechos. Se puede consultar el esquema de metadatos que ha implementado la Biblioteca Digital de California: http://www.cdlib.org/inside/projects/rights/schema/</p>

Unidad semántica	4.1.3.1 copyrightStatus
Componentes semánticos	Ninguno
Definición	Un código que designe el estado en que se encuentra el objeto conforme a los derechos de autor en el momento en que se registra.
Limitaciones	El valor debe proceder de un listado controlado de términos.
Repetible	No Repetible
Obligatoriedad	Obligatorio
Notas de creación/mantenimiento	Posibles valores: copyrighted = Sujeto a derechos de autor publicdomain = En dominio público. unknown = Se desconoce el estado del objeto en lo que respecta a los derechos de autor.

Unidad semántica	4.1.3.2 copyrightJurisdiction
Componentes semánticos	Ninguno
Definición	País de aplicación de la ley de derechos de autor.
Justificación	La ley de derechos de autor varía de un país a otro.
Limitaciones	El valor debe proceder del código ISO 3166
Ejemplo	us de
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.3.3 copyrightStatusDeterminationDate
Componentes semánticos	Ninguno
Definición	La fecha en que el estado de derecho de autor queda registrada en <i>copyrightStatus</i> .
Limitaciones	Este valor debería adoptar una forma estructurada para facilitar el proceso automatizado de los datos. Se recomienda utilizar convenciones normalizadas que faciliten el intercambio de metadatos conforme a PREMIS, por ejemplo las utilizadas en el esquema PREMIS.
Ejemplos	20070608
Repetible	No Repetible
Obligatoriedad	Opcional

Unidad semántica	4.1.3.4 copyrightNote
Componentes semánticos	Ninguno
Definición	Información adicional referente al estado de los derechos de autor del objeto.
Limitaciones	Ninguna
Repetible	Repetible
Obligatoriedad	Opcional
Ejemplos	Está previsto que los derechos de autor expiren en 2010 a no ser que se renueven. La declaración de los derechos de autor está en la cabecera del fichero.

Unidad semántica	4.1.4 licenseInformation
Componentes semánticos	4.1.4.1 licenseIdentifier 4.1.4.2 licenseTerms 4.1.4.3 licenseNote
Definición	Información acerca de una licencia u otro tipo de acuerdo que afecte a la concesión de permisos relacionados con un objeto.
Limitaciones	Contenedor
Repetible	No Repetible
Obligatoriedad	Opcional
Notas de uso	En los casos en los que el contenido del metadato <i>rightsBasis</i> sea "licencia", se debe cumplimentar esta información.

Unidad semántica	4.1.4.1 licenseIdentifier
Componentes semánticos	4.1.4.1.1 licenseIdentifierType 4.1.4.1.2 licenseIdentifierValue
Definición	Denominación utilizada para identificar de manera unívoca los términos de la concesión de permisos, dentro del repositorio.
Limitaciones	Contenedor
Repetible	No Repetible
Obligatoriedad	Opcional
Notas de uso	<p>Esta unidad semántica hace referencia al documento en el que se hace constar la concesión de permisos. En ocasiones puede ser el contrato final firmado por el cliente.</p> <p>El identificador es opcional puesto que los términos de la licencia pueden almacenarse fuera del repositorio.</p> <p>En algunos repositorios se utilizan acuerdos de concesión verbal, en este contexto, por ejemplo, la totalidad del acuerdo verbal puede incluirse en los <i>licenseTerms</i>.</p>

Unidad semántica	4.1.4.1.1 licenseIdentifierType
Componentes semánticos	Ninguno
Definición	Valor del <i>licenseIdentifier</i>
Limitaciones	Ninguno
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.4.1.2 licenseIdentifierValue
Componentes semánticos	Ninguno
Definición	Designación de dominio en el cual el identificador de la licencia es único.
Limitaciones	El valor debe proceder de un listado controlado de términos.
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.4.2 licenseTerms
Componentes semánticos	Ninguno
Definición	Texto que describe los términos de la licencia o el acuerdo verbal que garantiza los derechos de uso.
Limitaciones	Ninguno
Repetible	No Repetible
Obligatoriedad	Opcional
Notas de uso	Puede contener tanto el texto íntegro de la licencia como el acuerdo o bien un resumen.

Unidad semántica	4.1.4.3 licenseNote
Componentes semánticos	Ninguno
Definición	Información adicional referente a la licencia.
Limitaciones	Ninguno
Repetible	Repetible
Ejemplo	La licencia está embebida en el XMP de la cabecera del fichero.
Obligatoriedad	Opcional
Notas de uso	Toda la información referente a los términos de la licencia debe aparecer en <i>licenseTerms</i> . <i>licenseNotes</i> está destinado a albergar otro tipo de información relacionada con la licencia, por ejemplo, personas de contacto, o bien aclaraciones a la interpretación de sus términos. En nota también se puede indicar si los términos de la licencia están disponibles en línea o bien embebidos en el propio objeto.

Unidad semántica	4.1.5 statuteInformation
Componentes semánticos	4.1.5.1 statuteJurisdiction 4.1.5.2 statuteCitation 4.1.5.3 statuteInformationDeterminationDate 4.1.5.4 statuteNote
Definición	Información acerca del estatuto que permite el uso del objeto.
Limitaciones	Container
Repetible	Repetible
Obligatoriedad	Opcional
Notas de uso	Cuando el contenido de <i>rightsBasis</i> es "statute", el metadato <i>statuteInformation</i> debe cumplimentarse.

Unidad semántica	4.1.5.1 statuteJurisdiction
Componentes semánticos	Ninguno
Definición	País u otro órgano político en que el estatuto tiene aplicación.
Justificación	La conexión entre el objeto y los derechos concedidos se basan en la jurisdicción.
Limitaciones	Los valores deben proceder de un vocabulario controlado.
Ejemplo	Us De
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.5.2 statuteCitation
Componentes semánticos	Ninguno
Definición	Identificación del estatuto.
Limitaciones	Ninguno
Ejemplo	Legal Deposit (Jersey) Law 200- National Library of New Zealand (Te Puna Mātauranga o Aotearoa) Act 2003 no 19 part 4 s 34
Repetible	No Repetible
Obligatoriedad	Obligatorio
Notas de uso	Se deben citar de manera estandarizada en caso en los que dicha denominación exista.

Unidad semántica	4.1.5.3 statuteInformationDeterminationDate
Componentes semánticos	Ninguno
Definición	Fecha en la cual se resuelve aprobar el estatuto.
Justificación	El permiso en cuestión puede ser objeto de interpretación. El contenido de los estatutos depende de un contexto específico dado en un momento específico. En otras circunstancias, el contexto, y por lo tanto, el contenido del mismo, podría cambiar. Por esta razón, puede ser importante registrar la fecha de la aprobación del estatuto.
Limitaciones	Este valor debería adoptar una forma estructurada para facilitar el proceso automatizado de los datos. Se recomienda utilizar convenciones normalizadas que faciliten el intercambio de metadatos conforme a PREMIS, por ejemplo las utilizadas en el esquema PREMIS.
Ejemplo	2007-12-01 20040223151047.0
Repetible	No Repetible
Obligatoriedad	Opcional

Unidad semántica	4.1.5.4 statuteNote
Componentes semánticos	Ninguno
Definición	Información adicional acerca del estatuto.
Limitaciones	Ninguno
Ejemplo	Aplicable a todos los contenidos web remitidos para su revisión por consenso general 9/19/2008.
Repetible	Repetible
Obligatoriedad	Opcional

Unidad semántica	4.1.6 rightsGranted
Componentes semánticos	4.6.1 act 4.6.2 restriction 4.6.3 termOfGrant 4.6.4 rightsGrantedNote
Definición	Concesiones otorgadas al repositorio.
Limitaciones	Contenedor
Repetible	Repetible
Obligatoriedad	Opcional

Unidad semántica	4.1.6.1 act
Componentes semánticos	Ninguno
Definición	Acciones que el repositorio puede llevar a cabo.
Limitaciones	El valor debe proceder de un listado controlado de términos.
Repetible	No Repetible
Obligatoriedad	Obligatorio
Notas de uso	Posibles valores: Replicate = Proceso de crear una copia idéntica al original Migrate = proceso mediante el cual se realiza una copia idéntica en contenido pero, con un formato de fichero diferente Modify = realizar una versión con contenido diferente Use=lectura sin copia o modificación alguna Disemínate =creación de una copia o versión de uso fuera del repositorio Delete = proceso de eliminación de un objeto del depósito de almacenamiento
	Corresponde al repositorio decidir cuál debe de ser el nivel de especificidad del vocabulario controlado. Quizá sea de utilidad emplear los mismos valores se utilizan para eventType.

Unidad semántica	4.1.6.2 restriction
Componentes semánticos	Ninguno
Definición	Condiciones o limitaciones a la norma
Limitaciones	Ninguno
Ejemplo	No más de tres Disponibles tras un año de permanencia del objeto en el repositorio El titular de los derechos de autor deberá recibir una notificación tras finalizar el periodo de vigencia de la norma que afecta a su obra.
Repetible	Repetible
Obligatoriedad	Opcional

Unidad semántica	4.1.6.3 termOfGrant
Componentes semánticos	4.1.6.3.1 startDate 4.1.6.3.2 endDate
Definición	Plazo de vigencia de los permisos concedidos.
Justificación	Las concesiones pueden ser temporales.
Limitaciones	Container
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.6.3.1 startDate
Componentes semánticos	Ninguno
Definición	Fecha en que comienza la autorización del permiso.
Limitaciones	Este valor debería adoptar una forma estructurada para facilitar el proceso automatizado de los datos. Se recomienda utilizar convenciones normalizadas que faciliten el intercambio de metadatos conforme a PREMIS, por ejemplo las utilizadas en el esquema PREMIS.
Ejemplo	2006-01-02 20050723
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.6.3.2 endDate
Componentes semánticos	Ninguno
Definición	Fecha en que finaliza la autorización del permiso.
Limitaciones	Este valor debería adoptar una forma estructurada para facilitar el proceso automatizado de los datos. Se recomienda utilizar convenciones normalizadas que faciliten el intercambio de metadatos conforme a PREMIS, por ejemplo las utilizadas en el esquema PREMIS.
Ejemplo	2006-01-02 20050723
Repetible	No Repetible
Obligatoriedad	Opcional
Notas de uso	Se utiliza el término "OPEN" para indicar que se trata de un plazo abierto <i>endDate</i> se omite si se desconoce la fecha de finalización del permiso.

Unidad semántica	4.1.6.4 rightsGrantedNote
Componentes semánticos	Ninguno
Definición	Información adicional referente a los permisos concedidos.
Justificación	En ocasiones es necesario incluir una explicación adicional que clarifique los permisos concedidos.
Limitaciones	Ninguno
Repetible	Repetible
Obligatoriedad	Opcional
Notas de uso	En esta unidad semántica se pueden incluir datos referentes a evaluación de riesgos, por ejemplo, cuando un repositorio no está seguro acerca de qué permisos le han sido otorgados.

Unidad semántica	4.1.7 linkingObjectIdentifier
Componentes semánticos	4.1.7.1 linkingObjectIdentifierType 4.1.7.2 linkingObjectIdentifierValue
Definición	El identificador de un objeto relacionado con la declaración de derechos.
Justificación	Las declaraciones de derechos tienen que estar asociadas a los objetos a los que se refieren, ya sea mediante la vinculación de la declaración de derechos con el objeto(s) o bien por la vinculación del objeto(s) a la declaración de derechos.
Limitaciones	Container
Repetible	Repetible
Obligatoriedad	Opcional
Notas de uso	<i>linkingObjectIdentifier</i> es opcional. En algunos casos es más práctico vincular el objeto u objetos a una declaración de derechos. Por ejemplo, en aquellos casos en los que una misma declaración de derecho de dominio público afecta a los miles de objetos de un repositorio.

Unidad semántica	4.1.7.1 linkingObjectIdentifierType
Componentes semánticos	Ninguno
Definición	Designación del dominio en el cual el <i>linkingObjectIdentifier</i> es único.
Limitaciones	El valor debe proceder de un listado controlado de términos.
Ejemplo	[véanse los ejemplos para objectIdentifierType]
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.7.2 linkingObjectIdentifierValue
Componentes semánticos	Ninguno
Definición	Valor del <i>linkingObjectIdentifier</i>
Limitaciones	Ninguno
Ejemplo	[véanse los ejemplos para objectIdentifierValue]
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.8 linkingAgentIdentifier
Componentes semánticos	4.1.8.1 linkingAgentIdentifierType 4.1.8.2 linkingAgentIdentifierValue 4.1.8.3 linkingAgentRole
Definición	Identificador de uno o más agentes vinculados a la declaración de derechos.
Justificación	Las declaraciones de derechos pueden estar asociadas a los agentes bien mediante la vinculación de la declaración de derechos con el agente(s) o bien por la vinculación del agente(s) a la declaración de derechos.
Limitaciones	Container
Repetible	Repetible
Obligatoriedad	Opcional
Notas de uso	<i>linkingAgentIdentifier</i> es opcional, ya que un agente puede ser desconocido o irrelevante. Los agentes suelen ser irrelevantes cuando la base del derecho es un estatuto.

Unidad semántica	4.1.8.1 linkingAgentIdentifierType
Componentes semánticos	Ninguno
Definición	Designación del dominio en el cual el <i>linkingAgentIdentifier</i> es único.
Limitaciones	El valor debe proceder de un lenguaje controlado.
Ejemplo	[véanse los ejemplos para agentIdentifierType]
Repetible	Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.8.2 linkingAgentIdentifierValue
Componentes semánticos	Ninguno
Definición	El valor del <i>linkingAgentIdentifier</i> .
Limitaciones	Ninguno
Ejemplo	[véanse los ejemplos para agentIdentifierValue]
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.1.8.3 linkingAgentRole
Componentes semánticos	Ninguno
Definición	El rol del agente en relación a la declaración de derechos.
Limitaciones	El valor debe proceder de un listado controlado de términos.
Ejemplo	Contact= contacto Creator= Autor Publisher= Editor Rightsholder= Titular de los derechos Grantor= Responsable de la concesión de los derechos
Repetible	No Repetible
Obligatoriedad	Obligatorio

Unidad semántica	4.2 rightsExtension
Componentes semánticos	Definido externamente
Definición	Contenedor que incluye unidades semánticas definidas fuera del Diccionario de Datos PREMIS.
Justificación	En ocasiones puede ser necesario remplazar o ampliar unidades definidas en el Diccionario de Datos PREMIS.
Limitaciones	Contenedor
Repetible	Repetible
Obligatoriedad	Opcional
Notas de uso	<p>Se proporciona la <i>extensibilidad</i> con el fin de favorecer la granularidad y el uso de unidades semánticas definidas fuera del Diccionario. Se pueden incluir tanto unidades semánticas locales como metadatos procedentes de otro esquema de metadatos bien en adición bien en sustitución de las unidades semánticas definidas en el Diccionario PREMIS.</p> <p>Cuando se utilice un esquema de extensión, se debe proporcionar una referenciar al esquema utilizado. Para obtener más información relativa a la "extensibilidad" véase Capacidad de extensión.</p> <p>En caso de que la entidad Derecho esté incluida, <i>rightsStatement</i> o <i>rightsExtension</i> debe estar presente.</p> <p>Si el contenedor <i>rightsExtension</i> tiene que estar necesariamente asociado con alguna de las subunidades de derechos del Diccionario PREMIS, el contenedor Rights se repite. Si es necesario recurrir a la <i>extensibilidad</i> mediante esquemas de metadatos externos, el contenedor Rights también debe repetirse.</p>

3. TEMAS ESPECIALES

El grupo de trabajo PREMIS comprobó mientras redactaba el Diccionario de Datos que había una serie de temas importantes aunque demasiado específicos como para ser tratados en el Diccionario. Estos temas se exponen a continuación ya que proporcionan información esencial a cerca de las unidades semánticas y justifican las decisiones tomadas por el grupo.

3.1 Información de formato

El grupo de trabajo estudió el formato con detalle y llegó a la conclusión de que había que llegar a un acuerdo en algunos temas fundamentales antes de poder definir unidades semánticas concretas. Estos temas incluían:

- ¿Qué es un formato?
- ¿Qué tipo de objetos tienen formato?
- ¿Cómo se identifica un formato?
- ¿Existe alguna diferencia entre un formato y un perfil?

El concepto de formato parece casi intuitivo, pero debido a la importancia que tiene la información del formato para la preservación digital, el grupo decidió ser muy concreto respecto a su significado. Debatido acerca de las características que definen un formato surgió se llegó a la conclusión de que todo formato tiene que corresponderse con alguna especificación formal o informal, no puede tratarse de un diseño de bits al azar o sin previa documentación. La definición de Wikipedia, «una manera particular de codificar información para almacenarla en un archivo informático», no parece enfatizar lo suficiente esta característica²⁴. El grupo esbozó su propia definición: *una estructura específica y preestablecida para la organización de un fichero digital o cadena de bits.*

El formato es, evidentemente, una propiedad de los ficheros, pero también puede aplicarse a las cadenas de bits. Por ejemplo, una cadena de bits de una imagen dentro de un fichero TIFF podría tener un formato acorde a la especificación del formato del fichero TIFF. Por este motivo, PREMIS evita utilizar el término *formato de fichero* y emplea en su lugar *formato*, más genérico.

Un repositorio debe registrar la información sobre el formato de la manera más específica posible. Lo ideal sería identificar los formatos con un enlace directo hacia la especificación completa del formato. En la práctica, es más cómodo un enlace indirecto como un código o una cadena que pueda a su vez asociarse con las especificaciones completas del formato. El grupo consideró el nombre de formato como una designación algo arbitraria que se podría utilizar a modo de enlace indirecto. Sin embargo, surgieron dos complicaciones cuando el grupo intentó definir las unidades semánticas que se utilizarían como enlace.

En primer lugar, las designaciones utilizadas habitualmente para los formatos, como las del tipo MIME con sus extensiones, no ofrecen información suficientemente detallada como para utilizarla sin tener que añadir información adicional sobre la versión. Se debatió acerca de si la unidad semántica definida para el nombre del formato debería incluir tanto el formato como la versión (por ejemplo, «TIFF 6.0») o si deberían definirse dos unidades semánticas diferentes, una para el nombre y otra para la versión. Se decantaron por dos unidades semánticas para poder utilizar listados de autoridades ya existentes como los de tipo MIME, así, en el Diccionario de Datos, *formatDesignation* (designación del formato) consta de dos componentes: *formatName* (nombre de formato) y *formatVersion* (versión de formato).

En segundo lugar, es muy probable que en el futuro los registros centralizados de formatos constituyan la mejor vía para obtener información detallada sobre formatos²⁵. En el modelo PREMIS, el nombre del formato proporciona un enlace indirecto a la especificación del formato. En cuanto al registro hay que saber dos cosas: cuál es el registro que se utiliza y qué es lo que identifica la especificación dentro del registro.

El grupo se planteó o combinar todas las identificaciones de formato en un único conjunto de unidades semánticas, o bien definir contenedores diferentes según la existencia o no de registros.

Un buen argumento a favor del conjunto único es que un repositorio que utiliza su propia lista de autoridades de nombres de formatos para asociar los objetos digitales con sus especificaciones está básicamente manteniendo su propio registro de formatos, en el que la identificación del registro se asume sin más. Sin embargo, como todavía hay grandes registros de formatos en proceso de desarrollo el grupo, no se atrevió a aventurar que es lo que será necesario para utilizarlos. Finalmente,

se definieron dos contenedores: *formatDesignation* (designación de formato) y *formatRegistry* (registro de formato).

Dentro de un contenedor de formato, es obligatorio que al menos una de estas dos unidades semánticas esté presente para proporcionar la información identificativa necesaria. La relación es más evidente cuando se utilizan ambos.

El grupo decidió hacer *format* repetible para permitir los casos en los que a) se está utilizando más de un registro o en los que b) no es posible identificar el formato inmediatamente

- a) Si se utilizan varios registros, la repetición del elemento *format* permite registrar con claridad incoherencias entre los formatos identificados por cada uno de los registros. Para reducir la ambigüedad, debería utilizarse *formatRegistryRole* (función del registro de formatos) para indicar por qué razón se está utilizando el registro —por ejemplo, identificación de formato, validación de formato, caracterización o identificación del perfil—. Un registro concreto debería indicarse por *formatRegistryRole* como la fuente autorizada para identificar formatos. Debería utilizarse *formatNote* (nota sobre el formato) para registrar información adicional, por ejemplo, cuando varias identificaciones son verdaderas en conjunción [como en el caso de BWF y WAV].
- b) En la práctica, las aplicaciones para la identificación de ficheros pueden crear varias identificaciones posibles por fichero o cadena de bits, y resolver la identificación del formato puede no ser posible de manera inmediata. La repetición del elemento *format* hace posible capturarlos. Debería utilizarse *formatNote* (nota sobre el formato) para registrar información adicional cuando varias identificaciones hacen que haya una disyuntiva entre posibles formatos [por ejemplo, TIFF 3.0 o TIFF 4.0].

No es extraño que se especifique una determinada aplicación práctica de un formato, a menudo denominada perfil. Por ejemplo, GeoTIFF (para imágenes geográficas), TIFF/EP (para cámaras digitales) y TIFF/IT (para imágenes pre prensa) son compatibles con la especificación TIFF, pero la restringen al requerir ciertas opciones o la amplían al añadir etiquetas. Debido a esto, un fichero puede tener más de un formato, por ejemplo, tanto TIFF como GeoTIFF. El grupo valoró diversas opciones para adaptar esto, como hacer repetible la designación del formato o definir el perfil del formato como una unidad semántica diferente, pero

optaron por recomendar que en cada caso se registrara la más específica de las designaciones de formato. Con el fin de alcanzar este nivel más específico un repositorio (o un registro de formatos) puede utilizar nombres de formato compuestos (por ejemplo, «TIFF_GeoTIFF» o «WAVE_MPEG_BWF»).

El grupo asumió que puede haber diversidad de opiniones en cuanto a cuál es la designación más específica, y que dependerá de cada puesta en práctica. Por ejemplo, para un documento METS¹⁰ (es decir, un documento en XML que se ajuste al esquema METS), un repositorio puede considerar que XML es el formato más específico, mientras que otro puede considerar que es METS.

3.1.1 Entorno

Lo que diferencia a los materiales digitales de los analógicos es que en el caso de los digitales se interpone un entorno técnico complejo entre el usuario y el contenido. El software, los sistemas operativos, los recursos informáticos e incluso la conectividad de redes permiten al usuario recuperar el contenido e interactuar con él. Aislar el contenido digital de su contexto puede convertirlo en inutilizable, por lo tanto, la documentación exhaustiva sobre el entorno técnico de un objeto digital se convierte en un componente esencial de los metadatos de preservación.

Debido a que los entornos digitales están formados por componentes que se pueden dividir en componentes cada vez menores, es fácil que sus descripciones se vuelvan extremadamente complejas. También las descripciones pueden tender a ser las mismas para conjuntos de objetos digitales, por ejemplo, para todos los ficheros de un formato en particular. Estos dos factores sugieren que el modelo más eficaz para recoger y mantener los metadatos de entorno es un registro centralizado. Aunque el desarrollo del contenedor *environment* de PREMIS no suponía la existencia de este registro, puede entenderse como una plantilla para los tipos de información que un registro de entornos podría contener, más que para lo que se espera que el repositorio registre localmente.

Las unidades semánticas asociadas con el contenedor *environment* representan la recomendación del grupo de trabajo de PREMIS sobre qué es lo que un repositorio necesita conocer sobre el entorno de un objeto. El modo en el que se da a conocer esta información (a través de un registro central de metadatos almacenados localmente o a través de ambos) es un asunto que el repositorio deberá resolver durante la implementación.

El grupo de trabajo decidió limitar su ámbito a los metadatos sobre entorno asociados a los objetos que se encuentren actualmente en el repositorio. La estrategia para consignar los cambios en el entorno a lo largo del tiempo pertenece al ámbito de la implementación y, por lo tanto, queda fuera del ámbito teórico del Diccionario de Datos.

A veces un único formato digital funciona con varios entornos diferentes. El Diccionario de Datos admite esta posibilidad haciendo repetible el contenedor *environment*, pero no pretende que un repositorio deba intentar explicar cada combinación posible de software o hardware compatible con un determinado objeto almacenado. Sin embargo, los entornos documentados deberían incluir la unidad semántica *environmentCharacteristic* (características del entorno), con un valor apropiado como «mínimo», «recomendado», «operativo», etc. En general, el grupo de trabajo estaba de acuerdo con que se debería especificar, al menos, un entorno «mínimo». Las especificaciones de un entorno que es «operativo» pueden ser necesarias en los casos en que es importante mantener ciertas propiedades significativas del objeto: detalles del aspecto, comportamiento y funcionalidad originales del objeto. En estas circunstancias, es útil documentar un entorno en el que conste que funcionan estos atributos satisfactoriamente.

El grupo de trabajo se planteó si los metadatos de entorno podían aplicarse de forma útil a representaciones, ficheros o cadenas de bits, aunque en muchos casos no se aplican a las cadenas de bits, ya que el software opera con formatos de fichero conocidos o, en el caso de los objetos compuestos, con agrupaciones de formatos de fichero conocidos, se podrían aplicar a las cadenas de bits en algunas situaciones. Por ejemplo, es posible utilizar un único fichero AVI como contenedor común para distintos vídeo que requieran el uso de un software de reproducción diferente. En un fichero AVI que encapsule cadenas de bits heterogéneas, cada cadena de bits puede requerir un flujo de trabajo de preservación claramente único.

Al definir el entorno al nivel de las cadenas de bits se refuerza la idea de que una cadena de bits concreta requiere un entorno concreto. Si el entorno se estableciera al nivel de fichero, esta asociación se perdería, lo que complicaría los esfuerzos de preservación al requerir la descomposición del fichero.

Sin embargo, en otros casos, un formato de fichero puede contener dos o más cadenas de bits específicas con semánticas totalmente diferentes, pero el software establecido para soportar el formato puede interpretar correctamente o recuperar cualquier cadena de bits que aparezca en el fichero. Por ejemplo, un visor de

ficheros TIFF, al recuperar una imagen, se salta cierta información de la cabecera del fichero (una cadena de bits dentro del fichero) para llegar a los datos de la imagen (una segunda cadena de bits dentro del fichero). No siempre es necesario detallar por separado la información del entorno para cada una de estas cadenas de bits cuando ambas pueden recuperarse con una aplicación compatible con la especificación de formato TIFF.

Se debe tener en cuenta que los metadatos de entorno de un Objeto concreto pueden ser diferentes para la representación y para el fichero. Por ejemplo, un navegador es apropiado para mostrar una página web multimedia que conste de texto, imágenes fijas, animaciones y componentes de sonido; pero cada componente de la página web recuperado por separado requiere entornos específicos y diferentes a aquel que se utiliza para recuperar toda la página web en su conjunto.

El grupo de trabajo decidió no recomendar que se proporcionara distinta información de entorno para las versiones de preservación y difusión de un objeto (cuando la versión que se difunde es la versión que está disponible para los usuarios en un *Dissemination Information Package* o DIP, paquete de información para difusión). Si el repositorio guarda las versiones de difusión separadas de los modelos originales de preservación, éstos son objetos almacenados y pueden describirse mediante todos los metadatos aplicables a las entidades Objeto. Si las versiones de difusión se generan en tiempo real a partir de los modelos originales de preservación, el entorno correspondiente ya no atañe a la preservación. Aunque la información del entorno para las versiones de difusión pueda ser útil en algunos casos, no es esencial para los procesos de preservación. (Véase información sobre el [formato de difusión](#))

También se discutió la posibilidad de que el medio mediante por el que los objetos archivados se envían del repositorio al usuario (por ejemplo, a través de una red, en CD, en DVD, etc.) formara parte de los metadatos de entorno. El argumento a favor es que el entorno de reproducción debe ser compatible con los requisitos del medio de entrega: si el contenido se entrega en CD-ROM, el entorno de reproducción debe incluir un reproductor de CD-ROM. Sin embargo, el grupo decidió que no es esencial conocer el medio de entrega para el proceso de preservación y, por lo tanto, no se consideró fundamental. Además, la utilidad de describir un medio de entrega variará de un repositorio a otro, en función de las políticas locales de difusión.

Pese a la importancia fundamental que tienen los metadatos de entorno para garantizar que los materiales digitales permanezcan accesibles y utilizables durante mucho tiempo, el grupo de trabajo decidió, a su pesar, hacer optativo el contenedor *environment*. El grupo no pudo afirmar categóricamente que cada estrategia de preservación que existe o que se pueda desarrollar requiera conocer la información del entorno. Sin embargo, el hecho de que el contenedor *environment* sea actualmente optativo no significa que el grupo de trabajo no considere importante este metadato. Una adecuada documentación de los entornos en lo referente a su acceso y su uso son componentes esenciales de la mayoría de las estrategias de preservación digital. Aún queda mucho trabajo pendiente hasta llegar a establecer sistemas automáticos que recopilen, almacenen y actualicen este metadato.

3.1.2 Características y nivel de composición de los objetos: el modelo «cebolla»

Cuando se comprime o se cifra un objeto, el formato resultante depende del sistema de compresión o cifrado utilizado. Al mismo tiempo, el objeto tiene un formato subyacente que es diferente. Este tipo de objetos plantea el problema de cómo describir capas complejas de codificaciones y cifrados de manera que se puedan descodificar o descifrar correctamente. El grupo planteó la metáfora de una cebolla: un objeto digital puede estar envuelto en varias capas de codificación que necesiten «pelarse» siguiendo una secuencia concreta. El modelo «cebolla» considera cada capa como un nivel de composición y organiza los metadatos en conjuntos de valores, un conjunto por capa.

El ejemplo más simple es el de un fichero sin codificar ni cifrar. En este caso, habría una unidad semántica *objectCharacteristics* (características del objeto) con *compositionLevel* (nivel de composición) de valor 0 (cero). Las características de objeto de un PDF, por ejemplo, pueden incluir un mensaje cifrado, un tamaño de 500.000 bytes, un formato de PDF 1.2, inhibidores —por ejemplo, un inhibidor para impedir la impresión— y Adobe Acrobat como creador. Si se creara una versión comprimida de ese fichero PDF utilizando UNIX gzip y se almacenase en el repositorio, el fichero comprimido se describiría con dos bloques de *objectCharacteristics* (características de objeto). El primero, con *compositionLevel* (nivel de composición) cero, sería el mismo que el del PDF sencillo, y el segundo, con *compositionLevel* (nivel de composición) 1, registraría otro mensaje cifrado, más pequeño y con formato de gzip. Esto se aplicaría a tantas capas como fuera necesario para describir el objeto por completo.

Para extraer el objeto, se trabaja en sentido contrario a través de los niveles de composición, desde el mayor hasta el menor, utilizando una aplicación apropiada para el formato de la capa. En el ejemplo anterior, para conseguir el PDF se aplica una herramienta que entienda el formato gzip. Una vez descomprimido el contenido, se puede comparar con la información previamente almacenada sobre el tamaño y la fijeza para determinar si se ha extraído el objeto correcto (en la práctica, algunos codificadores tienen mecanismos de comprobación internos).

Cabe destacar que este modelo asume que el objeto se almacena con las capas de composición preservadas. Si el archivo ya ha eliminado las capas y está almacenando el objeto base, la información sobre la eliminación de las capas es información sobre Acontecimiento en vez de información sobre composición. Es decir, si se crea la versión descomprimida del objeto A y se llama objeto B, A está relacionado con B por una relación de derivación (sourceOf) con un acontecimiento de descompresión relacionado.

Las cadenas de bits y los ficheros no son capas de composición. Si un archivo elige gestionar objetos de cadenas de bits o flujos de ficheros, éstos son objetos diferentes cuyo lugar de almacenamiento se encuentra en el interior de un fichero, que a su vez es un objeto diferente con sus características y sus metadatos y con su propia ubicación. Cada uno de ellos puede tener capas de composición cifradas y codificadas. La capa de composición del nivel cero del archivo sería el fichero sin cifrar ni codificar, la cadena de bits dentro del fichero es un objeto en si mismo diferenciado de las capas de codificación del fichero.

Los formatos tar y ZIP, que permiten agrupar («empaquetar») varios ficheros en uno, presentan un problema parecido aunque no idéntico. Si el paquete consiste en un solo objeto, se puede considerar el paquete como otra capa de composición; por ejemplo, un fichero que está cifrado y después comprimido tendría tres niveles de composición. Sin embargo, si el paquete contiene más de un fichero, debería tratarse como un objeto independiente que comunica la ubicación de los objetos preservados, de tal manera que queden registros de metadatos diferentes para cada uno de los objetos preservados. Por ejemplo, un fichero ZIP que contenga dos ficheros PDF debería tratarse como tres objetos: el fichero ZIP con un formato de composición base de ZIP y otros dos objetos cuya ubicación está dentro del fichero ZIP. Como en las cadenas de bits, los objetos que están dentro de un objeto fichero ZIP son distintos del objeto contenedor. Cada uno puede tener conjuntos de metadatos completamente diferentes y puede presentar también capas de

composición adicionales. En un supuesto fichero ZIP cifrado que contuviera dos ficheros cifrados por separado, habría entonces tres objetos, cada uno con dos niveles de composición.

3.1.3 Fijeza, integridad y autenticidad

En el proceso de definición de los elementos fundamentales para la preservación, el grupo de trabajo prestó especial atención a los conceptos de fijeza, integridad y autenticidad de los objetos digitales. Los objetos que carecen de estas características tienen poco valor para los repositorios destinados a proteger el valor testimonial o, por supuesto, a conservar el patrimonio cultural.

En el Diccionario de Datos PREMIS, la información que se necesita para verificar la fijeza —es decir, la cualidad de un objeto para permanecer inalterado a partir de un momento anterior— se describe mediante un conjunto de componentes semánticos bajo la unidad semántica *objectCharacteristics* (características del objeto). La aplicación de un fixity check (programa de control de fijeza) sobre un objeto para detectar cambios no autorizados se detalla como un Acontecimiento. En el mundo analógico, son la publicación y la producción de las obras las que fijan el objeto. En el contexto digital, pueden utilizarse los algoritmos de *hash* que crean la información resumen en forma de mensaje cifrado (*message digest*) para llevar a cabo un control de la fijeza de un objeto. Si el mensaje cifrado que se ha creado a través de un algoritmo en un momento dado es idéntico al mensaje cifrado creado por el mismo algoritmo posteriormente, esto indica que el objeto no ha sufrido ningún cambio durante el período intermedio. De hecho, la práctica recomendada es crear y comprobar al menos dos mensajes cifrados utilizando dos algoritmos diferentes para asegurar la fijeza del objeto.

Aunque este proceso puede indicar con cierta seguridad que un objeto no ha cambiado con el tiempo, no indica nada acerca de la integridad o autenticidad del objeto. En el modelo PREMIS, la integridad de un objeto se verifica mediante un Acontecimiento. La identificación y validación del formato son indicadores fundamentales de la integridad de un fichero. Mediante tecnología informática, como la de JHOVE, se puede comprobar que la extensión del fichero coincida efectivamente con su formato y determinar el nivel de conformidad con unas especificaciones de formato determinadas²⁶. La integridad de una representación se debe verificar con programas especiales que entiendan la estructura de la representación. Si la representación incluye metadatos estructurales, estos se

pueden utilizar para comprobar que todos los ficheros estén presentes y tengan el nombre adecuado.

La autenticidad de un objeto digital es la cualidad de ser lo que pretende ser. Como explica la Coalición para la Preservación Digital (Digital Preservation Coalition, DPC) «En el caso de los registros electrónicos, [la autenticidad] se refiere a la fiabilidad del registro electrónico en tanto que registro (...) La cuestión de la confianza en la autenticidad de los materiales digitales a lo largo del tiempo es especialmente importante debido a la facilidad con que se pueden realizar alteraciones²⁷».

La autenticación o demostración de autenticidad tiene varias facetas e incluye aspectos tanto técnicos como de procedimiento. Desde el punto de vista técnico se puede incluir el mantenimiento de la documentación detallada acerca de la procedencia digital (la historia del objeto), la preservación de una versión del objeto que, en lo que respecta a los bits, es idéntica al contenido, y el uso de firmas digitales.

Los metadatos PREMIS son capaces de documentar la procedencia mediante la definición de las unidades semánticas asociadas con Acontecimientos y mediante los enlaces entre las entidades Objeto y las entidades Acontecimiento. Se puede contrastar la fijeza con la información del mensaje cifrado almacenado y registrar la propia comprobación como un Acontecimiento.

3.1.4 Firmas digitales

Los repositorios de preservación utilizan la firma digital de tres maneras:

- Para la **entrega** al repositorio. Un agente (autor o remitente) puede firmar un objeto para demostrar que es el verdadero autor o remitente.
- Para la **difusión**. El repositorio puede firmar un objeto para demostrar que, efectivamente, es la auténtica fuente de la difusión.
- Para el **almacenamiento en archivos**. Un repositorio puede archivar objetos firmados de manera que sea posible confirmar el origen y la integridad de los datos.

El primero y el segundo son de uso común hoy en día, ya que las firmas digitales se utilizan en la transmisión de documentación tanto comercial como de otro tipo. Generalmente, la validación tiene lugar poco después de la firma y no hace falta conservarla. En el primer caso, el repositorio puede registrar el acta de validación

como un Acontecimiento y guardar como detalles de éste la información relacionada que se precisa para demostrar la procedencia. En el segundo caso, el repositorio también puede registrar la firma como un Acontecimiento, pero su uso es responsabilidad del receptor. Sólo en el tercer caso, en el que el repositorio utiliza las firmas digitales como una herramienta para confirmar la autenticidad de sus objetos digitales almacenados a lo largo del tiempo, se debe preservar la propia firma y la información necesaria para validarla.

Al igual que ocurre con las firmas físicas o los sellos, las firmas digitales fidedignas requieren que:

- La firma sea exclusiva del firmante.
- La firma ataña al contenido del documento que se ha firmado.
- La firma pueda ser reconocida por terceros como la firma de la persona o entidad que la ha realizado.

Para crear una firma digital, primero se aplica al contenido (un fichero o una cadena de bits) un algoritmo de *hash* seguro (Secure Hash Algorithm) y se utiliza para crear un mensaje cifrado corto a partir del contenido. El mensaje cifrado y, opcionalmente, la información relacionada, se codifican utilizando una criptografía asimétrica. La criptografía asimétrica se basa en el uso de dos claves: una clave privada para cifrar y una clave pública para descifrar. El firmante debe custodiar la clave privada de forma secreta y a ser posible en un soporte físico seguro. Con esto se consigue que la firma sea exclusiva de la persona que rubrica. Al relacionar directamente el mensaje cifrado con el contenido, la firma queda así automáticamente relacionada también con éste. La firma se puede verificar descifrándola con la clave pública del firmante y comparando el mensaje resumen ya descifrado con un nuevo mensaje producido por el mismo algoritmo a partir del mismo contenido. Si el contenido hubiera cambiado, el resultado de la comparación no sería el mismo.

El objetivo de relacionar la firma con el firmante se basa en el establecimiento de la confianza. Por ejemplo, el agente A debería confiar en la firma del agente B si un tercero en quien confía A afirma que la firma pertenece realmente a B. Por este principio se rige la certificación ante notario de las firmas escritas. Lo mismo ocurre con las firmas digitales, donde un tercero de confianza certifica que una clave en concreto es la clave pública del firmante. Esto se extiende a lo largo de una cadena de confianza, por la que el agente en quien se confía, a su vez confía en un

intermediario, quien por su parte certifica la clave pública del firmante. Por lo general, este proceso suele ejecutarse utilizando los certificados X.509 o cadenas de certificados.

Esta cuestión es importante en el campo de la preservación, pues los mecanismos normalizados actuales para establecer la confianza en un certificado son responsabilidad de unos servicios que probablemente no sigan estando disponibles en el futuro. Compartir la clave pública de manera generalizada y almacenarla de forma segura como un documento formal puede resultar un enfoque apropiado para la preservación. Por ejemplo, una universidad podría publicar con regularidad sus claves públicas en sus anuarios y ponerlas a disposición del público en su página web.

Metadatos de firma digital

Para que un repositorio de preservación pueda validar una firma digital, deberá almacenar lo siguiente:

- La propia firma digital
- El nombre del algoritmo de *hash* y del algoritmo de cifrado utilizados para producir la firma digital
- Los parámetros asociados a estos algoritmos
- La cadena de certificados que se necesita para validar la firma (si se utiliza un modelo de certificado para relacionar al firmante con su clave pública)

Se recomienda que el repositorio también almacene las definiciones de los algoritmos y las normas relevantes (por ejemplo, para codificar las claves) de manera que estos métodos se puedan volver a ejecutar si fuera necesario.

El sistema XML-Signature Syntax and Processing (XMLDsig) de W3C actúa como la norma de hecho para codificar las firmas digitales y proporciona un modelo funcional claro²⁸. PREMIS adoptó los nombres y la estructura de las unidades semánticas de esta especificación en los casos en los que se podía utilizar. Sin embargo, *XMLDsig* es demasiado general y, por otra parte, demasiado específico para poder aplicarse de manera universal en un contexto de preservación. Es demasiado general porque permite que múltiples objetos de datos (ficheros o cadenas de bits en el modelo PREMIS) se firmen juntos, mientras que en el modelo PREMIS la firma digital es propiedad de un único objeto. Además, es demasiado

específico porque exige una codificación y una metodología de validación determinadas que no se pueden aplicar de manera universal.

El Diccionario de Datos define la siguiente estructura:

1.9 signatureInformation (información sobre la firma) (M, R) [fichero, cadena de bits]

1.9.1 signature (firma) (M, R)

1.9.1.1 signatureEncoding (codificación de la firma) (O, NR) [fichero, cadena de bits]

1.9.1.2 signer (firmante) (M, NR) [fichero, cadena de bits]

1.9.1.3 signatureMethod (método de la firma) (O, NR) [fichero, cadena de bits]

1.9.1.4 signatureValue (valor de la firma) (O, NR) [fichero, cadena de bits]

1.9.1.5 signatureValidationRules (reglas de validación de la firma) (O, NR) [fichero, cadena de bits]

1.9.1.6 signatureProperties (propiedades de la firma) (M, R) [fichero, cadena de bits]

1.9.1.7 keyInformation (información sobre la clave) (M, NR) [fichero, cadena de bits]

1.9.2 signatureInformationExtension (extensión de la información sobre la firma) (M, R) [fichero, cadena de bits]

Los algoritmos de *hash* y de cifrado utilizados se registran en el *signatureMethod* (método de la firma); por ejemplo, «DSA-SHA1» indicaría que el algoritmo de cifrado es DSA y que el algoritmo de *hash* es SHA1.

La propia firma digital es el *signatureValue* (valor de la firma). La información sobre la generación de la firma que no se necesita para validarla (por ejemplo, la fecha y la hora en que se creó la firma) se guarda en *signatureProperties* (propiedades de la firma). La clave pública utilizada para validar la firma se indica en *keyInformation* (información sobre la clave). Dado que hay muchos tipos de claves, cada una con

estructuras diferentes, estas estructuras no se definieron en el Diccionario de Datos, y las personas encargadas de ponerlo en práctica necesitarán utilizar estructuras definidas externamente. Por eso, *keyInformation* (información sobre la clave) se define como un contenedor extensible. Los repositorios pueden utilizar las definiciones «*KeyInfo*» (información sobre la clave) cuando sea necesario.

Las unidades semánticas que se han abordado anteriormente tienen versiones análogas en el *XMLDsig*:

PREMIS	XMLDsig
signatureMethod (método de la firma)	<SignedInfo><SignatureMethod> (información firmada, método de la firma)
signatureValue (valor de la firma)	<SignatureValue> (valor de la firma)
signatureProperties (propiedades de la firma)	<Object><SignatureProperties> (objeto, propiedades de la firma)

Tres unidades semánticas que no estaban incluidas en *XMLDsig* se han añadido al Diccionario de Datos:

signatureEncoding (codificación de la firma), *signer* (firmante), y *signatureValidationRules* (reglas de validación de la firma). La unidad semántica *signatureEncoding* (codificación de la firma) indica la codificación de los valores de las unidades semánticas siguientes, lo cual no está incluido en *XMLDsig* porque ese documento exige una determinada codificación que no se puede asumir para un contexto más amplio. El nombre del firmante puede extraerse de su certificado si se incluye en la *keyInformation* (información de la clave), pero aislar esta información en *signer* (firmante) facilita el acceso. La documentación del proceso que se ha de seguir en la validación de la firma se almacena en *signatureValidationRules* (reglas de validación de la firma). Como en el caso de *signatureEncoding* (codificación de la firma), esto no se halla en *XMLDsig* porque *XMLDsig* requiere un método de validación concreto.

En los casos en los que el repositorio pueda y prefiera utilizar *XMLDsig*, se puede usar el esquema completo en lugar del contenedor *signature* de PREMIS mediante el contenedor de extensión *signatureInformationExtension* (extensión de la información sobre la firma). En este caso, los elementos obligatorios en PREMIS o

bien son obligatorios en *XMLDsig* (signatureMethod [método de la firma], signatureValue [valor de la firma]) o están implícitos en los requisitos de la especificación *XMLDsig* (signatureEncoding, [codificación de la firma] signatureValidationRules [reglas de validación de la firma]). En los casos en los que un repositorio no pueda usar *XMLDsig* u opte por no usarlo, todavía podrá usar los elementos «KeyInfo» definidos en el esquema *XMLDsig* para definir la unidad semántica registrada en *keyInformation* (información sobre la clave).

3.1.5 Metadatos no fundamentales

Aunque el grupo de trabajo decidió no incluir algunas definiciones de metadatos en el Diccionario de Datos, no significa que en otros contextos estas unidades semánticas no sean necesarias o importantes. Para determinadas implementaciones podría justificarse el registro de esta información.

Agrupación: la inserción de objetos dentro de otros mayores (en lugar de una colección de objetos separados). Una agrupación se distingue por la presencia de múltiples ficheros o cadenas de bits, que se documentarán en *objectCharacteristics* (características del objeto). Esta unidad semántica no distingue entre una agrupación que entra por un proceso de ingest y una agrupación que se crea a través del repositorio de preservación para almacenarse o para cualquier otro propósito. Sin embargo, esta distinción no se considera fundamental.

Anomalías y peculiaridades: El *Framework* define *peculiaridades* como «toda pérdida en la funcionalidad o cambio en la apariencia o percepción del objeto de datos de contenido a resultas de los procesos de preservación y los procedimientos ejecutados por el archivo». El grupo de trabajo utilizó el término *anomalías* para designar los aspectos de un objeto que no concuerden con sus especificaciones propias. El debate sobre las peculiaridades y las anomalías se centró en si debían definirse como resultados de Acontecimientos o clasificarse como propiedades de Objetos.

El argumento a favor de tratarlos como resultados de acontecimientos se basa en que las peculiaridades, por definición, surgen de un acontecimiento, y las anomalías se descubren a través del acontecimiento de validación. Considerada de este modo, una anomalía podría registrarse como parte de la descripción de un acontecimiento de validación; la unidad semántica *eventOutcome* (consecuencias del acontecimiento) daría indicación de problemas y la unidad semántica

eventOutcomeDetail (detalles sobre las consecuencias del acontecimiento) registraría las anomalías conocidas.

Una razón para tratar las peculiaridades y las anomalías como propiedades de un objeto consiste en que esto parece concederles más importancia y les proporciona una asociación directa con el objeto, en lugar de una indirecta.

La decisión es arbitraria. El Diccionario de Datos considera las peculiaridades y anomalías como consecuencias de los acontecimientos, registrados en *eventOutcomeDetail* (detalles sobre las consecuencias del acontecimiento).

Codificación de caracteres: Este elemento es importante, pero se trata de metadatos técnicos específicos que dependen del formato, que sólo son útiles para ficheros de texto y ficheros que incluyan texto.

Conformidad con el perfil: Un *perfil* puede entenderse como un subtipo o como un formato refinado; por ejemplo, la especificación GeoTIFF se puede considerar como un perfil de TIFF. Se planteó la cuestión sobre si la conformidad con el perfil se debía ver como algo independiente de la validación del formato. La decisión de que se recomendara registrar en cada caso la más específica de las designaciones de formato obvió la necesidad de definir una unidad semántica independiente para la conformidad con el perfil.

Fecha de modificación: El modelo de datos PREMIS afirma que los metadatos sólo describen un objeto en un momento determinado. Si un objeto cambia o se modifica, se crea uno nuevo relacionado con el anterior. Por lo tanto, cada objeto tiene su propio conjunto de metadatos y la relación entre los dos también se describe. El modelo no permite modificar un objeto y mantener un conjunto de metadatos que describan un historial con sus cambios. Por lo tanto, no existiría una fecha de modificación del objeto, sino solamente una fecha de creación para el nuevo objeto. El acto de modificación (por ejemplo, migración o normalización) se documenta como un Acontecimiento y se enlaza con el objeto creado como resultado de estos procesos. La fecha de modificación fue considerada por el grupo como el registro de un Acontecimiento asociado a un Objeto, y no como una fecha asociada al historial de cambios de los metadatos asociados al Objeto.

Formato de difusión: Se discutió mucho acerca de si el formato de difusión se incluía dentro del ámbito de trabajo. El grupo de trabajo acordó que el objeto de la Preservación es el "formato de preservación" que puede o no coincidir con el de

difusión. El que el formato de preservación esté disponible inmediatamente o sea transformado en el momento de la difusión es una decisión que se toma en el proceso de puesta en práctica. Por ejemplo, si el formato de preservación es una imagen TIFF, un repositorio de preservación podría crear una versión de difusión (una imagen JPG) sobre la marcha para el acceso por parte del usuario, mientras que otro repositorio podría enviar el TIFF original. Un tercer repositorio puede almacenar y procesar tanto el TIFF original como la copia de acceso JPEG.

El Diccionario de Datos no estudia la creación de objetos de metadatos que no estén almacenados en un repositorio de preservación. Aunque el grupo coincidía en que el formato de difusión es importante para el repositorio en el ámbito operativo, no es fundamental para los procesos de preservación.

Identificador Global: el *Framework* incluía un "Global Identifier" (identificador global) definido como un identificador conocido fuera del sistema del repositorio. El grupo no consideró importante distinguir entre un identificador conocido externamente y uno conocido internamente. Un identificador interno podría darse a conocer fácilmente fuera del repositorio y convertirse así en un identificador global. La cuestión era saber si los identificadores internos serían lo suficientemente inequívocos en un contexto externo para funcionar como identificadores globales. Sin embargo, como el *objectIdentifier* (identificador del objeto) siempre incluye un tipo de identificador además del valor, esta combinación de tipo y valor sería exclusiva, incluso si el tipo fuera el de algún esquema de repositorio local.

El *Framework* también entendía que un identificador global sería un identificador estándar, del tipo ISBN o ISSN. Sin embargo, debido a que estos esquemas designan una entidad bibliográfica abstracta o un conjunto de elementos, y no los objetos de datos específicos del repositorio de preservación, son en realidad metadatos descriptivos en vez de metadatos de preservación. Los identificadores estándar, como ISBN, ISSN y otros similares, suelen referirse a representaciones muy diferentes contenidas en muchos repositorios de preservación distintos, sin posibilidad de distinguir entre ellos. Por lo tanto, el identificador empleado por el repositorio debe ser, en la práctica, el identificador "global".

Metadatos embebidos: uno de los supuestos prácticos incluyó un indicador de metadatos para señalar si un objeto fichero contenía metadatos embebidos. El grupo acordó dejar este indicador fuera del Diccionario de Datos por el momento, a sabiendas de que esto probablemente se debería revisar en los próximos años con

el aumento gradual de formatos que incluyen metadatos embebidos. Actualmente, si se extrae un metadato embebido y se almacena en cualquier otro lugar, no es necesario señalar la existencia de metadatos embebidos en el fichero.

El grupo también abordó el tema de la distinción entre los metadatos embebidos estándar definidos por un formato de fichero y los metadatos definidos localmente que se pueden insertar en la cabecera de un fichero. Toda divergencia local de los formatos estándar seguramente necesitará documentarse como anomalía.

Motivo de la creación: este elemento se definió en el *Framework*. El grupo de trabajo llegó a la conclusión de que, para los objetos creados por el repositorio de preservación (por ejemplo, una versión normalizada de un fichero), el motivo de la creación podría registrarse como parte de *eventDetail* (detalle del acontecimiento) para el acontecimiento de creación.

Sin embargo, el grupo no consideró en profundidad los acontecimientos o procesos que ocurren antes del proceso de ingest y no estaba convencido de que esta información fuera fundamental para el repositorio de preservación. El contexto que engloba la creación de objetos puede documentarse parcialmente mediante la Entidad Objeto en *creatingApplication* (aplicación creadora). El grupo expresó algunas reservas acerca del modelo de ciclo de vida utilizado por el *Framework* (origen, preingesta, ingesta, retención en archivos, etc.) por considerarlo demasiado restrictivo.

Niveles de permanencia: el grupo estudió el modo en que los índices de permanencia de la National Library of Medicine estaban relacionados con el trabajo de PREMIS²⁹. El índice del nivel de permanencia parece ser una propiedad de las reglas de negocio (business rules) más que una propiedad de la entidad Objeto. El grupo ya había decidido excluir las reglas de negocio (business rules) del ámbito del Diccionario de Datos.

Nombre de la ruta del fichero/URI: se consideró que este elemento era tanto específico de la implementación como dependiente del sistema. No se consideró que tuviera que quedar registrado en un repositorio.

A menudo, la ruta o ubicación de un objeto no se conoce en un sistema de gestión de contenido; sólo se conoce y se necesita para la recuperación el identificador exclusivo de objeto. Alternativamente, en algunos sistemas, como Handle, el *objectIdentifier* (identificador de objeto) por sí solo suele ser suficiente para

recuperar el fichero. Por lo tanto, se definió una unidad semántica menos dependiente del sistema y más amplia: *contentLocation* (localización del contenido). Se puede interpretar de forma limitada (un valor puede ser una ruta exacta o una ruta «totalmente cualificada» o un nombre de fichero) o bien de una manera más amplia (cualquier información que se precise para recuperar un fichero de un sistema de almacenamiento, que puede incluir información utilizada por un sistema de resolución como el sistema Handle).

Orden de Bytes: determina si los números de más de ocho bits se guardan de más a menos significativos (*big-endian*) o de menos a más significativos (*little endian*). El orden de bytes depende del hardware y puede causar problemas cuando los datos se comparten entre diferentes tipos de ordenadores. Sin embargo, no afecta a todos los formatos. Por ejemplo, es intrascendente en las codificaciones como ASCII, donde un byte equivale a un carácter, o como UTF-8, que es ajena al orden de bytes. El grupo de trabajo decidió que sería mejor que el orden de bytes se tratara como metadatos técnicos para cada formato específico, y destacó que ANSI/NISO Z39.87 (Diccionario de Datos – *Technical Metadata for Digital Still Images*¹²) incluye el orden de bytes como un metadato técnico para imágenes.

Programación del acontecimiento: Muchas acciones del repositorio de preservación se llevan a cabo periódicamente, por ejemplo, las acciones de control diario o semanal. Puede ser útil registrar la fecha de la acción o activar una alarma que indique cual es el siguiente acontecimiento programado. Esto se consideró parte de la política del repositorio y de la implementación, y no como una propiedad fundamental de los Acontecimientos.

Relación de hermandad: el grupo debatió acerca de si las relaciones entre hermanos (“hijos” del mismo padre) debían constituir una categoría de relación independiente. Se acordó que las relaciones de hermandad siempre tendrían una relación estructural (y también podrían tener una relación de derivación) y, por lo tanto, deberían englobarse en la categoría de esas relaciones. Lo que las hace potencialmente confusas es el hecho de que el padre no siempre se guarda dentro del sistema del repositorio. Por ejemplo, a partir de un informe creado con Microsoft Word podría crearse una versión PDF para imprimir y una versión HTML para publicar online. Si estas dos representaciones se almacenaran en el archivo de preservación sin el fichero original en Word, no sería obvio que las dos representaciones tuviesen una relación de hermanos.

Tipo de acontecimiento: la unidad semántica *eventType* (tipo de acontecimiento) se considera fundamental, pero no es así con todos los tipos de acontecimientos, y algunos se omitieron intencionadamente en la lista de valores sugeridos en el Diccionario de Datos. Entre ellos, el grupo coincidió en que la microfilmación (cambio de formato del material preservado), la extracción de un fichero de la red interna y la actualización de soportes no eran acontecimientos fundamentales. Los acontecimientos que se pudieran gestionar en un sistema de almacenamiento, como la creación de copias espejo o de copias de seguridad, se grabarían en un registro del sistema y no se elevarían a la categoría de acontecimientos con metadatos asociados.

Tipo de objeto: el grupo estudió si era deseable contar con una unidad semántica para un género o un tipo de soporte que clasificara los objetos a un nivel más alto que el de formato. Hay un elemento así en el esquema METS, pero actualmente no dispone de vocabulario controlado. El grupo planteó que el tipo de objeto es información útil en el sistema (por ejemplo, para llevar a cabo la preservación de una clase de materiales) y, posiblemente, a la hora de clasificar en función de su forma de reproducción en ciertos entornos. La tipificación de objeto a alto nivel es probablemente más útil para el intercambio y el acceso que para la preservación. Sin embargo, desarrollar una lista de tipos de objeto universalmente aceptada está fuera del ámbito de PREMIS y, sin una lista autorizada de tipos, este elemento no sería útil fuera del repositorio. Este elemento se puede registrar en los metadatos descriptivos.

Tipo MIME: Internet Media Type and SubType (conocidos como «tipo MIME») se incluyeron en *formatIdentification* (identificación del formato). La identificación del formato pretende ser más precisa y con mayor nivel de detalle que el tipo MIME e incluye múltiples esquemas de identificación de formato, entre los que puede incluirse el tipo MIME. Un tipo MIME por sí solo no es lo suficientemente riguroso para identificar formatos para la preservación digital por varias razones: no todos los formatos son del tipo MIME, se trata de un mecanismo demasiado tosco, no está actualizado y no proporciona información de la versión. La práctica adecuada consiste en incluir el nombre de formato y la versión y utilizar el tipo MIME sólo si no hay otros datos disponibles.

4. METODOLOGÍA

El Subgrupo para Elementos Fundamentales (Core Elements Subgroup) comenzó por analizar las recomendaciones contenidas en Preservation Description Information (Información Descriptiva sobre Preservación) hechas por el anterior grupo de trabajo, Preservation Metadata Framework. El modelo OAIS, Preservation Description Information incluye información de referencia (identificadores e información bibliográfica), información de contexto (cómo se relacionan los objetos entre ellos), información de procedencia (la historia del contenido digital) e información de fijeza. Los miembros del subgrupo que pertenecían a instituciones que utilizaban o desarrollaban repositorios de preservación mapearon elementos del *Framework* con elementos utilizados en sus propios sistemas. El subgrupo también revisó especificaciones de organizaciones y proyectos que no tenían representación en el grupo de trabajo PREMIS.

Quedó claro que los elementos tipo detallados en el *Framework* no siempre correspondían a los elementos utilizados en la práctica. Sin embargo, el ejercicio proporcionó un denominador común para varias implementaciones; el grupo estudió cada elemento en diversas audioconferencias para descubrir aspectos comunes en su uso. Los elementos con un uso más extendido formaron la base para un conjunto de elementos fundamentales que fueron asignados a los tipos de entidad apropiados según evolucionaba el modelo de datos.

Tanto el modelo OAIS como el *Framework*, consideran los metadatos técnicos Representation Information (información de la representación) más que Preservation Description Information. Dado que el *Framework* recoge pocos metadatos técnicos, el grupo de trabajo compiló una lista de posibles metadatos técnicos basados en especificaciones propuestas para Global Digital Format Registry (Registro de Formatos digital global o GDFR), y además añadió elementos utilizados en repositorios de las instituciones participantes³⁰. Cada elemento de la lista se estudió detalladamente, y todos los que fueran especificaciones de algún formato o implementaciones concretas se consideraron no fundamentales. En ocasiones se consultó a expertos en otras áreas, especialmente complejas, como las relativas a formatos, información sobre entornos de hardware y software y firmas digitales.

Se analizaron y estudiaron elementos que provenían de distintas fuentes para decidir si se podían aplicar y determinar finalmente cuáles eran las unidades semánticas fundamentales.

El grupo de trabajo, como norma general excluyó del Diccionario de Datos:

- Elementos de metadatos que pudieran agruparse en categorías más amplias.
- Elementos específicos de algún formato, o propios de una implementación o fruto de políticas concretas de actuación.
- Elementos fuera del ámbito de PREMIS
- Elementos cuya información se pudiera obtener de manera fácil y fiable a partir del propio objeto o de otras fuentes.

5. GLOSARIO

El grupo de trabajo de PREMIS pronto identificó la necesidad de contar con un vocabulario aceptado sobre metadatos de preservación. A continuación se definen algunos de los términos utilizados en este informe. El grupo de trabajo asume que, otros especialistas hayan podido definir algunos de estos términos de manera diferente. Los vocablos incluidos en el glosario se seleccionaron en función de su importancia, de la frecuencia con la que aparecen en el informe y en el Diccionario de Datos, o por eliminar ambigüedades en su interpretación.

Los términos subrayados en el Glosario cuentan con su propia definición dentro del mismo.

Accionable: propiedad de una [Unidad Semántica](#) que indica que está registrada o codificada de manera que se pueda procesar automáticamente.

Acontecimiento: acción que afecta por lo menos a un [Objeto digital](#) y/o a un [Agente](#) conocidos por el [Repositorio de preservación](#).

Actualización: véase Actualización de soporte.

Actualización de soporte: procedimiento por la cual un [Objeto digital](#) se copia a otra unidad de almacenamiento que puede utilizar un tipo de soporte igual o similar. Observación: utilizamos [Actualización de soporte](#) y no simplemente

actualización, a fin de evitar el significado que atribuye el modelo OAIS a este término. El modelo OAIS define actualización como una "Migración digital por la cual se sustituye una unidad de soporte por una copia lo suficientemente exacta que permita que el hardware de almacenamiento de archivos pueda seguir ejecutándose tal como lo hacía".

Agente: actor (persona, máquina o software) asociado a uno o más [Acontecimientos](#) relacionados con un [Objeto digital](#).

Almacenar: guardar un [Fichero](#) en un dispositivo de almacenamiento permanente como un disco, una cinta o un DVD.

Análisis de virus informáticos (Virus Check): proceso que consiste en examinar un [Fichero](#) en busca de programas maliciosos diseñados para corromper sistemas y Objetos Digitales.

Anomalía: aspecto de un [Objeto digital](#) que no concuerda con su especificación.

Autenticidad: propiedad que indica si un [Objeto digital](#) es lo que lo que declara ser; es decir que la integridad de la fuente y del contenido del [Objeto digital](#) se pueden verificar.

Business Rules: véase: [Reglas de negocio](#)

Byte: componente de la jerarquía de datos de la máquina, normalmente mayor que un bit y menor que una palabra. Actualmente, un byte se compone de 8 bits que es la unidad de almacenamiento más pequeña. Un byte contiene normalmente un carácter. (Traducción de un extracto de FOLDOC: foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=byte)

Cadena de bits: datos contiguos o no contiguos dentro de un [Fichero](#) que presentan propiedades comunes a efectos de preservación. Una cadena de bits no puede convertirse en un fichero autónomo sin añadir una estructura de [Fichero](#) (cabeceras, etc.) ni reformatear la cadena de bits para adaptarse a un formato concreto. Esta definición es más concreta que las definiciones genéricas de cadena de bits (bitstream) que se utilizan en informática.

Cadena de ficheros: cadenas embebidas de bits que pueden convertirse en [ficheros](#) autónomos sin necesidad de incluir información adicional. Por ejemplo, una imagen TIFF incrustada en un fichero tar, o un EPS codificado dentro de un fichero XML.

Cálculo del mensaje cifrado: proceso mediante el cual se crea un mensaje [cifrado](#) para un [Objeto digital](#) que reside en un [Repositorio de preservación](#). Véase también [Comprobación de fijeza](#).

Capacidad de extensión: propiedad por la cual las [Unidades semánticas](#) del Diccionario de Datos PREMIS pueden complementarse con [Unidades semánticas](#) definidas externamente o sustituirse por otras que presenten mayor nivel de detalle, siempre que no existan conflictos en su definición y uso.

Captura: proceso mediante el cual un [Repositorio de preservación](#) capta [Objetos digitales](#) para su preservación a largo plazo. Por ejemplo, mediante un programa que recopila [Páginas web](#). El proceso de captura precede al proceso de [Carga de objetos digitales \(ingest\)](#).

[Carga de objetos digitales \(ingest\):](#) proceso que consiste en añadir objetos al sistema de almacenamiento de un [Repositorio de preservación](#). En el contexto del modelo OAIS, el proceso de ingest comprende servicios y funciones que son necesarios para aceptar paquetes de información (Submission Information Packages o SIP) procedentes de autores, editores, etc, y los transforma en uno o más paquetes de información (Archival Information Packages o AIP) para su preservación a largo plazo.

Cifrado: proceso que consisten en utilizar cualquier procedimiento criptográfico para transformar texto plano en texto cifrado ([mensaje codificado](#) o encriptado) para impedir que nadie más que el destinatario del mensaje pueda leer los datos. Esquemáticamente, existen dos clases de cifrado: la criptografía de clave pública y la criptografía de clave privada; por lo general, se usan de manera complementaria. Un algoritmo de criptografía de clave pública es el RSA; algunos algoritmos de clave privada son el Data Encryption Standard, en desuso y el Advanced Encryption Standard, así como el RC4. (Traducción de un extracto de FOLDOC: foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=encryption)

Componente semántico: Unidad semántica agrupada con una o más Unidades semánticas en un Contenedor. Un componente semántico puede constituir en sí mismo un Contenedor.

Compresión: proceso de codificación de datos para ahorrar espacio de almacenamiento o tiempo de transmisión. Aunque los datos ya estén codificados de forma digital para su procesamiento informático, a menudo todavía se pueden codificar más (utilizando menos bits). Por ejemplo, la codificación run-length

sustituye cadenas de caracteres repetidos (u otras unidades de datos) por una única aparición del carácter y un recuento de apariciones. Existen abundantes utilidades y algoritmos de compresión. Los datos comprimidos se deben [Descomprimir](#) para poder usarlos. (Traducción de un extracto de FOLDOC: foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=compression)

Comprobación de fijeza (Fixity Check): proceso para verificar que un [Fichero](#) o [Cadena de bits](#) no ha cambiado durante un período determinado. Un método habitual de comprobación de fijeza consiste en computar un mensaje cifrado o hash en un momento concreto y recalcularlo más tarde; si los resultados son idénticos, el objeto no se ha modificado.

Contenedor: en el Diccionario de Datos, una [Unidad semántica](#) utilizada para agrupar otras [Unidades semánticas](#) relacionadas. Una unidad semántica contenedora carece de valor por sí sola.

Derechos: declaración de derechos o permisos adquiridos por un [Objeto digital](#) y/o un [Agente](#).

Descifre: resultado de un procedimiento criptográfico que transforma texto cifrado (datos codificados o encriptados) en texto plano. (Traducción de un extracto de FOLDOC: foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?decryption)

Descompresión: proceso de invertir los efectos de la compresión de datos. (Traducción de un extracto de FOLDOC: foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?decompress)

Difusión: proceso de recuperación de un [Objeto digital](#) de un [Repositorio de preservación](#) para ponerlo a disposición de los usuarios. En el contexto del modelo OAIS, la Difusión implica transformar un paquete (o más) de información (Archival Information Package o AIP) en un paquete de información para difusión (Dissemination Information Package o DIP) y hacerlo accesible en un formato acorde a las necesidades de la comunidad de usuarios designada.

Duplicar: proceso que consiste en copiar un [Objeto digital](#) de modo que la copia resultante sea idéntica, en cuanto a bits, al objeto original. La [Migración de soporte](#) y la [Actualización de soporte](#) son dos clases concretas de este proceso.

Emulación: estrategia de preservación utilizada para solventar la obsolescencia tecnológica de hardware y software mediante el desarrollo de técnicas que imiten el funcionamiento de sistemas antiguos en generaciones modernas de ordenadores.

(Traducción de un extracto de DPC:
www.dpconline.org/graphics/intro/definitions.html)

Entidad: concepto abstracto que se aplica a un conjunto de realidades (agentes, acontecimientos, etc.) que comparten idénticas propiedades. El modelo de datos PREMIS define cinco tipos de entidades: [Entidades intelectuales](#), Objetos, [Agentes](#), [Derechos](#) y [Acontecimientos](#).

Entidad intelectual: conjunto de datos de contenido coherente que se considera como una unidad, por ejemplo, un libro, un mapa, una fotografía, una serie. Una Entidad intelectual puede contener otras Entidades intelectuales; por ejemplo, un sitio web puede contener una página web; una página web puede contener una fotografía. Una Entidad intelectual puede tener una o varias Representaciones.

Espacio de nombres (namespace): un grupo de nombres en el que todos son únicos, ninguno se repite. (Traducción de un extracto de FOLDOC:
foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?namespace)

Expurgo: proceso de retirada de un [objeto digital](#) del inventario de un [Repositorio](#).

Fichero: secuencia de [bytes](#) nombrada y ordenada que puede ser reconocida por un sistema operativo. Un fichero puede contener cero o más bytes, tiene permisos de acceso y contiene datos relativos al sistema de ficheros, tales como tamaño y última fecha de modificación. Un Fichero también tiene formato.

Fichero de datos: véase [Fichero](#).

Fijeza: propiedad que indica que un Objeto Digital no ha sido modificado en un lapso de tiempo concreto.

Fixity Check: véase [Comprobación de Fijeza](#).

Firma digital: valor computado con un algoritmo criptográfico y agregado a los datos de tal manera que el destinatario de los datos pueda usar la firma para verificar el origen y la integridad de dichos datos.

Es el equivalente electrónico a una firma manuscrita en un documento impreso. (Traducción de un extracto de BBN Technologies: www.bbn.com/utility/glossary/D)

Formato: estructura específica y preestablecida para la organización de un [Fichero](#), una [Cadena de bits](#) o una [Cadena de ficheros](#).

Inhibidor: característica de un [Objeto digital](#) cuyo propósito es restringir el acceso, copia, divulgación o migración. Entre los inhibidores más habituales destacan: la encriptación y la protección con contraseña.

Mensaje cifrado (Message digest): resultado de aplicar una función hash unidireccional a un mensaje. Un Mensaje cifrado (message digest) es un valor más corto que un mensaje, pero que sería distinto si el mensaje cambiara aunque sólo fuera en un carácter. (Traducción de un extracto de BBN Technologies: www.bbn.com/utility/glossary/M). Por mensaje se entiende cualquier sucesión de bits, como un fichero o cadena de bits. Un mensaje cifrado a menudo se conoce informalmente con el nombre de checksum o suma de verificación.

Message digest: véase: [Mensaje cifrado](#)

Metadatos de preservación: información de apoyo que utiliza un [Repositorio de preservación](#) en el proceso de preservación digital.

Metadatos de preservación fundamentales: [Unidades semánticas](#) que la mayoría de los [Repositorios de preservación](#) necesitarán conocer para soportar el proceso de preservación digital. Los metadatos de preservación fundamentales deberían ser independientes de factores como una estrategia concreta de preservación, o tipos de contenido archivado o políticas institucionales.

Metadatos descriptivos: metadatos que ayudan a la recuperación (cómo encontrar un recurso), la identificación (cómo distinguir un recurso de otros) y la selección (cómo establecer si un recurso responde a una determinada necesidad. Por ejemplo, la versión en DVD de una grabación en vídeo). (Traducción de un extracto de Caplan, Metadata Fundamentals for All Librarians, ALA Editions, 2003)

Metadatos estructurales: reflejan la estructura interna de los recursos digitales y las relaciones entre sus partes. Se utilizan para permitir la navegación y la presentación. (Traducción de un extracto de la NINCH Guide to Good Practice [guía de buenas prácticas NINCH]: www.nyu.edu/its/humanities/ninchguide/appendices/metadata.html)

Metadatos técnicos: información que describe los atributos físicos (en contraposición a los intelectuales) o las propiedades de los [Objetos digitales](#). Algunas propiedades de metadatos técnicos son específicas de cada [formato](#) (es decir, se aplican únicamente a [Objetos digitales](#) con un formato concreto; por ejemplo, el espacio de colores asociado a una imagen TIFF), mientras que otros son

independientes del Formato (es decir, se aplican a todos los Objetos digitales independientemente del Formato, por ejemplo, el tamaño en bytes).

Migración: estrategia de preservación mediante la cual una [Transformación](#) crea una versión de un [Objeto digital](#) en un [Formato](#) distinto, siendo el nuevo [Formato](#) compatible con entornos de software y hardware actuales. Lo ideal es que la Migración se complete con la mínima pérdida de contenido, formato y funcionalidad posible, pero la cantidad de pérdida de información variará en función de los Formatos y de los tipos de contenido afectados. También se llama migración de formato y migración hacia delante.

Observación: Se prefiere el uso de Migración y Migración de soporte al concepto de migración digital del modelo OAIS. OAIS define migración digital como la "transferencia de información digital, en un intento por preservarla, dentro de OAIS. Se distingue de las transferencias en general por tres atributos: 1) prestan especial atención a la preservación del contenido 2) consideran las nuevas incorporaciones de información en los archivos como sustituciones; y 3) creen que el control y la responsabilidad final sobre todos los aspectos de la transferencia residen en el OAIS".

Migración de formato: véase [Migración](#).

Migración de soporte (refreshing): procedimiento por el cual un [Objeto digital](#) se copia a un soporte de almacenamiento digital diferente ya que el medio digital original corre riesgo de obsolescencia.

Migración digital: véase [Migración](#).

Migración hacia delante: véase [Migración](#).

Normalización: forma de [Migración](#) por la cual se crea una versión de un [Objeto digital](#) en un nuevo [Formato](#), con propiedades más adecuadas para el tratamiento de la preservación. La Normalización a menudo se ejecuta como parte del proceso de [Carga de objetos \(ingest\)](#).

Namespace: véase: [Espacio de nombres](#)

Objeto: véase [Objeto digital](#).

Objeto complejo: véase [Objeto compuesto](#).

Objeto compuesto: objeto digital formado por varios [Ficheros](#), por ejemplo, una [página web](#) compuesta por ficheros de texto y de imagen.

Objetos de datos: véase [Objeto digital](#).

Objeto digital: unidad de información específica en formato digital. Un Objeto digital puede ser una [representación](#), un [fichero](#), una [cadena de bits](#) o una [cadena de ficheros](#). La definición de PREMIS para [Objeto digital](#) difiere de la definición comúnmente utilizada en el campo de las bibliotecas digitales, según la cual un Objeto digital es una combinación de identificador, metadatos y datos.

Objeto simple: [Objeto digital](#) que consta de un único [fichero](#), por ejemplo, un informe técnico completo en PDF.

Página web: página perteneciente a World Wide Web, generalmente en formato HTML/XHTML (las extensiones suelen ser .htm o .html) y que contiene enlaces de hipertexto que permiten navegar de una página o sección a otra. Las páginas web a menudo tienen asociados Ficheros gráficos para proporcionar ilustraciones que, a su vez, también pueden constituir enlaces. (Traducción de la definición inglesa extraída de Wikipedia: en.wikipedia.org/wiki/Web_page; entrada en español: es.wikipedia.org/wiki/Página_web)

Peculiaridad: pérdida de funcionalidad o cambio en la apariencia o percepción de un [Objeto digital](#) como resultado de los procesos y Procedimientos de preservación ejecutados por un repositorio de preservación. (Véase también la definición ofrecida por la Biblioteca Nacional de Australia: www.nla.gov.au/preserve/pmeta.html#14)

Perfil: especificación para la implementación concreta de un [Formato](#). Por ejemplo, GeoTIFF es un perfil de TIFF.

Permiso: acuerdo entre un titular de derechos y un [Repositorio de preservación](#) que permite a éste llevar a cabo ciertas acciones.

Preservación a escala de bits: estrategia de preservación en la que el único objetivo es garantizar que un [Objeto digital](#) permanezca fijo (inalterado) y viable (legible desde un soporte). No se hace ningún esfuerzo para garantizar que el [Objeto digital](#) siga siendo recuperable por la tecnología actual.

Procedencia digital: documentación relativa a los procesos que han tenido lugar a lo largo del ciclo de vida de un [Objeto digital](#). La procedencia normalmente describe a los [Agentes](#) responsables de la custodia y administración de objetos digitales y a los acontecimientos clave que suceden durante el curso de la vida de un Objeto digital. Describe asimismo otros datos asociados a la creación, gestión y preservación de los Objetos digitales.

Raíz: Fichero que se debe procesar en primer lugar para reproducir correctamente una representación.

Refreshing: véase Migración de soporte

Reglas de negocio (Business Rules): políticas y otras restricciones, guías y procedimientos que rigen la administración y el funcionamiento de un [Repositorio de preservación](#).

Relación: declaración sobre una asociación entre [Entidades](#).

Relación de dependencia: relación en la cual un Objeto digital requiere de otro [Objeto digital](#) para desempeñar su función o servicio, o para la coherencia de su contenido.

Relación de derivación: relación entre [Objetos digitales](#) en la que un objeto es el resultado de una transformación realizada en otro.

Relación estructural: relación entre las partes de un [Objeto digital](#).

Repositorio de Preservación: repositorio que, bien como responsabilidad única o como una de varias responsabilidades, tiene a cargo la preservación a largo plazo de [Objetos digitales](#).

Representación: [Objeto digital](#) que materializa o representa una [Entidad intelectual](#). Una representación es el conjunto de [Ficheros](#) y metadatos estructurales almacenados y necesarios para conseguir una reproducción completa y con sentido de una [Entidad intelectual](#).

Reproducir: hacer que un Objeto digital sea perceptible para un usuario, por medio de la visualización (en materiales audiovisuales), la reproducción (en materiales de audio) u otros medios adecuados al formato de un Objeto digital.

Sitio web: conjunto de páginas web, es decir, documentos HTML/XHTML accesibles a través del protocolo HTTP de Internet; todos los sitios web accesibles públicamente que existen conforman la World Wide Web.

A las páginas de un sitio web se accede desde una URL raíz común llamado portada o página de inicio (home page), que normalmente reside en el mismo servidor físico. Las URL organizan las páginas en una jerarquía, aunque los hiperenlaces entre ellas controlan cómo el lector percibe la estructura general y cómo el tráfico web fluye entre las diferentes partes de los sitios. (Adaptado de Wikipedia: es.wikipedia.org/wiki/Sitio_web)

Suma de verificación: véase Mensaje cifrado.

Supresión: eliminación de un Objeto digital de un repositorio.

Transformación: proceso aplicado a un Objeto digital que tiene como resultado uno o más Objetos digitales nuevos que no son idénticos en cuanto a bits al Objeto digital de origen. Algunos ejemplos de transformación son la Migración y la Normalización.

Unidad semántica: propiedad de una [Entidad](#). Observación: El Diccionario de Datos PREMIS establece una distinción entre [Unidad semántica](#) y metadato. Una unidad semántica consta de información que un [Repositorio de preservación](#) necesita conocer. El metadato es el registro de esa información. De este modo, en la práctica podría haber una relación de uno a uno entre unidad semántica y su metadato asociado, una relación de uno a varios, o incluso una relación de varios a uno.

En última instancia, el mapeo de un conjunto de unidades semánticas con su correspondiente conjunto de metadatos depende de cada implementación.

Validación: proceso por el que se coteja un [Objeto digital](#) con un estándar o parámetro de referencia y se observan las coincidencias o excepciones al respecto. Por ejemplo, un fichero puede validarse usando una especificación o perfil de formato de fichero; una [Representación](#) puede validarse con criterios de integridad.

Validación de firma digital: proceso mediante el cual se determina que una firma digital codificada se corresponde con un valor esperado cuando se introducen los algoritmos, claves y parámetros correctos. La validación confirma el autor y la fijeza del objeto digital firmado.

Viabilidad: propiedad de ser legible.

Virus Check: véase: [Análisis de virus informáticos](#).

NOTAS

- ¹ A Metadata Framework to Support the Preservation of Digital Objects (Dublin, Ohio: OCLC Online Computer Library Center, 2002),
http://www.oclc.org/research/activities/past/orprojects/pmwg/pm_framework.pdf
- ² Implementing Preservation Repositories for Digital Materials: Current Practice and Emerging Trends in the Cultural Heritage Community (Dublin, Ohio: OCLC Online Computer Library Center, 2004),
<http://www.oclc.org/research/activities/past/orprojects/pmwg/surveyreport.pdf>
- ³ Reference Model for an Open Archival Information System (OAIS) (Washington, DC: Consultative Committee for Space Data Systems, 2002),
<http://public.ccsds.org/publications/archive/650x0b1.pdf>
- ⁴ Otras iniciativas de metadatos de preservación han desarrollado modelos diferentes. La Biblioteca Nacional de Nueva Zelanda define cuatro tipos de entidades: objetos, archivos, procesos, y modificación de metadatos. Metadata Standards Framework—Preservation Metadata (Revised) (Wellington: National Library of New Zealand, June 2003), <http://www.natlib.govt.nz/catalogues/library-documents/preservation-metadata-revised>
- ⁵ Debe tenerse en cuenta que la definición de PREMIS para Objeto digital difiere de la definición comúnmente utilizada por la comunidad científica especializada en bibliotecas digitales, según la cual un Objeto digital es una combinación de identificador, metadatos y datos. La entidad Objeto en el modelo PREMIS es una abstracción que se define con el fin de agrupar atributos (unidades semánticas) y para mostrar relaciones.
- ⁶ IFLA, Functional Requirements for Bibliographic Records (Munich: K.G. Saur, 1998),
<http://www.ifla.org/en/publications/functional-requirements-for-bibliographic-records>
- ⁷ Coyle, Karen, Rights in the PREMIS Data Model,
<http://www.loc.gov/standards/premis/Rights-in-the-PREMIS-Data-Model.pdf>
- ⁸ Hirtle, Peter B., Digital Preservation and Copyright,
http://fairuse.stanford.edu/commentary_and_analysis/2003_11_hirtle.html

- ⁹ California Digital Library, copyrightMD schema,
<http://www.cdlib.org/inside/projects/rights/schema/>
- ¹⁰ Metadata Encoding & Transmission Standard (METS),
<http://www.loc.gov/standards/mets/>
- ¹¹ The Dublin Core Metadata Element Set, <http://www.dublincore.org/documents/dces/>.
- ¹² Data Dictionary—Technical Metadata for Digital Still Images, ANSI/NISO Z39.87-2006, <http://www.niso.org/standards>
- ¹³ Information technology – Multimedia framework (MPEG-21), ISO/IEC 21000 (multiple parts), International Organization for Standardization.
- ¹⁴ Synchronized Multimedia Integration Language (SMIL), <http://www.w3.org/TR/REC-smil/>
- ¹⁵ Resource Description Framework (RDF), <http://www.w3.org/RDF/>
- ¹⁶ MARC 21, <http://www.loc.gov/marc/>
- ¹⁷ Metadata Object Description Schema (MODS), <http://www.loc.gov/standards/mods/>
- ¹⁸ Content Standard for Digital Geospatial Metadata, FGDC-STD-001-1998,
<http://www.fgdc.gov/metadata/csdgm/>
- ¹⁹ VRA Core 4.0, <http://www.vraweb.org/projects/vracore4/>
- ²⁰ Encoded Archival Description (EAD), <http://www.loc.gov/ead/>
- ²¹ Data Documentation Initiative (DDI), <http://www.ddialliance.org/>
- ²² vCard, <http://www.imc.org/pdi/>
- ²³ Metadata Authority Description Schema (MADS),
<http://www.loc.gov/standards/mads/>
- ²⁴ Wikipedia, la enciclopedia gratuita disponible en: en.wikipedia.org/wiki/Main_Page
- ²⁵ Véase por ejemplo: Global Digital Format Registry disponible en:
<http://www.qdfr.info/>
- ²⁶ JHOVE – JSTOR/Harvard Object Validation Environment,
<http://hul.harvard.edu/jhove/>
- ²⁷ Digital Preservation Coalition Handbook,

<http://www.dpconline.org/advice/preservationhandbook>

²⁸ XML-Signature Syntax and Processing, W3C Recommendation 12 February 2002,
<http://www.w3.org/TR/xmlsig-core/>

²⁹ Margaret Byrnes, Assigning Permanence Levels to NLM's Electronic Publications
(presentado en: 2000 Preservation: An International Conference on the
Preservation and Long Term Accessibility of Digital Materials),

http://library.oclc.org/cdm4/item_viewer.php?CISOROOT=/p267701coll33&CISOPTR=134

³⁰ Global Digital Format Registry (GDFR) Data Model,

<http://www.gdfr.info/docs.html>