

Libro blanco de las tecnologías de documentación de LuCAS

**Descripción, funcionamiento, herramientas y futuro del
proyecto de documentación LuCAS**

Libro blanco de las tecnologías de documentación de LuCAS: Descripción, funcionamiento, herramientas y futuro del proyecto de documentación LuCAS

Publicado \$Id: libro-blanco-tecnologias-doc.sgml,v 1.3 2001/06/06 17:41:56 olea Exp \$

Tabla de contenidos

Prefacio	9
LuCAS desarrollo.....	11
Introducción	13
I. LuCAS v3.*	15
LuCAS: un proyecto de documentación libre en español	15
La libertad del acceso a la información	17
Objetivos	17
Orígenes	17
Breve cronología	17
Evolución	18
Funciones	18
Tipos de documentos	18
Desarrollo cooperativo.....	20
La Comunidad LuCAS	21
Metodología y herramientas del web LuCAS	21
Cifras	22
Licencias de documentación abierta	23
Desarrollos futuros	23
Glosario	24
Bibliografía	25
Creación en LuCAS: Ejemplos.....	27
Introducción	29
Proyectos de traducción o revisión	29
Integración de proyecto externo.....	31
Cómo conseguir el éxito y la fama	33
LuCAS v3.5.....	35
OBJETIVOS.....	37
FASE 1.....	37
FASE 2.....	37
FASE 3.....	37
Web de LuCAS: Directrices de diseño.....	39
Objetivos	41
Estado actual de la web	41
Nuevo web: principio general	41
Características del nuevo web	41
II. LuCAS v4 o las tecnologías necesarias para un proyecto de documentación moderno	43
Arquitectura de desarrollo de proyectos de documentación libre (Diseño e implementación de LuCAS v4).....	43
Antecedentes	45
Necesidades de un proyecto de documentación de soft libre.....	45
Técnicas que resuelven los objetivos	45
Herramientas que resuelven los objetivos	46
Sistema de publicación y consulta distribuido de nueva generación:	
Donantonio.	46
Formato estructurado para los documentos.....	46
Libro de estilo.....	47
Otros recursos lingüísticos	48
Taller de creación y mantenimiento de documentos: TallerDoc.....	48
Generación automática de formatos de publicación: Impresor-e.....	50
Publicación de documentos libres en Internet	51
Definiciones	53
Enfoque de este artículo	53

Documento transparente y documento opaco	53
La publicación web y la publicación de documentos.....	53
Solución al problema de la publicación web de documentos	53
Cómo debe publicarse un documento-e (libro-e) «LIBRE»	53
Documentos estructurados	55
Introducción	57
Ventajas de la estructuración de documentos	57
Herramientas que asistan a la creación	57
Referencias sobre docbook	57
Recursos lingüísticos: herramientas lingüísticas para el autor.....	59
Introducción	61
El asistente de coordinación de LuCAS v4.....	63
Introducción	65
Descripción de TallerDoc	67
Introducción	69
Sistema de escritura cooperativa.....	69
El impresor-e de LuCAS v4	71
taca	73
Introducción a Donantonio.....	75
Introducción	77
Especificación de requisitos de Donantonio.....	77
Introducción a Donantonio.....	79
Requisitos.....	81
Especificaciones técnicas	81
¿Especificación funcional?	82
Sistema de consultas	83
Sistema de replicación de recursos.....	84
Esquemas bibliográficos	84
Consecuencias prácticas de la arquitectura Donantonio	85
Herramientas a desarrollar.....	86
Terminología Donantonio	86
La metáfora Donantonio.....	87
Quién es Don Antonio	87
Bibliografía	87
Sistema distribuido automático de publicación. (era "Donantonio, an universal indexer architecture (draft)").....	91
Sistema automático de publicación.....	93
Sistema automático de publicación.....	93
Introducción: al principio era LuCAS.....	94
Requirements for an automatic distributed publishing architecture.....	94
Architecture versus application.....	94
Donantonio vocabulary	95
The Donantonio metaphor	96
Who's Don Antonio?.....	98
Change-log.....	98
Donantonio: sistema bibliográfico de publicación distribuida automática.....	99
Introducción	101
Descripción general.....	103
Requisitos específicos.....	104
Apéndices	107
Servidores Donantonio.....	109
Introducción	111
Descripción general.....	113
Apéndices	117
Clientes Donantonio	119
Introducción	121

Descripción general.....	123
Requisitos específicos.....	125
Apéndices	127
Interfaz de usuario Donantonio	129
Introducción	131
Descripción general.....	133
Requisitos específicos.....	134
Apéndices	137
Un día en la vida de un recurso en la red Donantonio.....	139
Aplicaciones de donantonio	143
Dorsales.....	145
Replicación de recursos	145
Tolerancia a caídas de la red	145

Prefacio

Aviso para esta edición: Este documento está en pleno desarrollo. Faltan numerosas secciones, otras están sólo perfiladas, otras partes deben ser reescritas y tal vez lleguen a reestructurarse otras.

Este libro pretende plasmar en su primera parte toda la experiencia acumulada proyecto LuCAS y en la segunda definir el proyecto tecnológico a seguir. En cualquier caso es el reflejo del trabajo hecho durante meses por un equipo excepcional y que no ha sido apenas divulgado.

Los éxitos, los logros y los resultados de los proyectos de documentación libres han sido tremendos y en algunos casos espectaculares. Sin embargo hay demasiados problemas y complejidades técnicas (en lo informático, editorial, lingüístico y ortotipográfico) que desde nuestro punto de vista tienen solución si se le dedica un esfuerzo razonable.

Este libro documenta nuestras propuestas y soluciones y la implementación de las mismas en herramientas.

En cuanto al idioma usado en el libro, que se haya usado español puede suponer un error para muchos, especialmente para aquellos que no sepan leer la lengua de Nebrija. Dada la absoluta generalidad y aplicabilidad de estas tecnologías en cualquier lengua es obvio lo conveniente de estar disponible en otros idiomas. Así que, aunque el origen tuvo lugar en el seno de un proyecto de documentación en español y que apoyamos expresamente el desarrollo de la ciencia y la tecnología en español, se favorecerá la disponibilidad del documento en inglés e incluso de la ampliación del mismo en inglés o español sin más reparo.

Finalmente queremos pedir a la comunidad de activistas en documentación, recursos lingüísticos, lingüística computacional y programadores que se tome la molestia de examinar este documento y las ideas que alberga para proponer las mejoras precisas y especialmente para implementarlas y darles uso. Sea como sea, todos compartimos la misma responsabilidad de servicio y calidad y pensamos que estas propuestas tecnológicas son el complemento adecuado a los esfuerzos de todos.

Prefacio

LuCAS desarrollo

El esfuerzo de investigación y desarrollo de recursos lingüísticos del proyecto LuCAS

Ismael Olea

Tabla de contenidos

Introducción.....	13
-------------------	----

Introducción

Clama al cielo la necesidad de mejorar las herramientas y técnicas de creación y gestión de documentación, especialmente si queremos que sean libres (osea, de libre uso, copia y modificación). Así se constituye en LuCAS un esfuerzo de desarrollo y creación de recursos lingüísticos.

En este breve artículo se citan los recursos disponibles actualmente.

- La documentación del proyecto a través del propio *Libro blanco de las tecnologías de documentación de LuCAS*.
- El servidor cvs
- La lista de correo lucas-desarrollo.

LuCAS: un proyecto de documentación libre en español

Juan José Amor

David Escorial

Ismael Olea

Palabras clave en este documento son: LuCAS, GNU/Linux, LDP, LDP-es, documentación en español, software libre, biblioteca electrónica.

Tabla de contenidos

La libertad del acceso a la información	17
Objetivos	17
Orígenes.....	17
Breve cronología.....	17
Evolución.....	18
Funciones.....	18
Tipos de documentos	18
Desarrollo cooperativo.....	20
La Comunidad LuCAS.....	21
Metodología y herramientas del web LuCAS.....	21
Cifras	22
Licencias de documentación abierta	23
Desarrollos futuros	23
Glosario.....	24
Bibliografía	25

La libertad del acceso a la información

Una condición imprescindible a la hora de publicar un documento es que éste sea «libre». Inspirados por la ética filosófica propuesta por la FSF creemos que la documentación debe estar al acceso de todo el mundo. Así su uso y copia debe ser completamente libre. También queremos ofrecer acceso a las fuentes, bien para ampliar o mejorar los trabajos o bien para permitir futuras transformaciones a otros formatos. Esta es una forma de crear una importante base de conocimiento para toda la comunidad.

Objetivos

Hablando en términos de ingeniería del software:

Producto Acabado = Programa Software + Manual de Usuario

- La calidad de un producto software se ve mermada si carece de la adecuada documentación.
- Producto no documentado = poco útil al usuario.

Por tanto se busca hacer llegar a todos el software libre con suficientes garantías de comprensión (textos de referencia).

Si bien el origen primero del proyecto era el de la traducción de las guías del LDP, exclusivamente dedicadas a Linux, el desarrollo del proyecto ha evolucionado hacia la creación, traducción y publicación de documentos dedicados a GNU/Linux, *IX y software libre en general.

Orígenes

LuCAS (LinUx en CASTellano) es un proyecto documental nacido como reflejo del LDP (Linux Document Project) y que junto a otros forma lo que se puede llamar LDP-es.

Aunque tuvo su origen físico en España (1995), desde el principio tuvo vocación internacional. Al ampliar el ámbito de publicación desde la red Fidonet hasta Internet ese carácter de internacionalidad ha ido creciendo hasta el nivel alcanzado hoy día.

Breve cronología

- 1995: Traducción de «Linux Instalación y Primeros Pasos»
- 1995-97: Llegada de otras guías: «Guía de Administración de Redes», «Guía del Usuario» ...
- 1995-97: Documentos cortos (CÓMOs)
- 1997-2000: Producción plena
 - Tutoriales, guías, CÓMOs, páginas de manual...
 - No sólo de Linux o software GNU (ej: licencias)

- Herramientas: glosario, diccionarios...

Evolución

A lo largo de la existencia de LuCAS se pueden identificar las siguientes fases:

- V1, proyecto de traducción y publicación de una única guía.
- V2, traducción y publicación de guías LDP y publicación (vía replicación) de todo el material disponible entonces (CÓMOS).
- V3, publicación (vía replicación) de todos los proyectos y esfuerzos nuevos y antiguos (centralización de la recopilación de información) y creación de una red de réplicas del sitio principal (distribución del acceso a la información); todo ello es asistido por automatismos de mantenimiento.
- V4, en desarrollo: arquitectura de desarrollo de proyectos de documentación libre

Funciones

En la versión actual de LuCAS (V3) se desarrollan las siguientes funciones que están enumeradas sin orden concreto:

- Creación y asistencia a la creación de documentos de todo tipo (traducciones y originales).
- Mantenimiento de una lista de correo de coordinación y apoyo a la creación.
- Replicación y apoyo de un Glosario de terminología informática inglés/español.
- Publicación de documentos recogidos por la *Red* que sin haber sido creados desde el grupo son afines a los objetivos del proyecto.
- Servicio de publicación para otros proyectos de documentación o localización al español.
- Servicio de replicación mundial de los mismos.

La división de labores no es estricta. Así es normal que algunos proyectos (documentos) surjan en alguno de los foros que forman el LDP-es y luego, en función de las metas del autor y de la clase de documento, acaben en otro más adecuado. Así los contenidos van quedando agrupados en función de su tipo facilitando las búsquedas de información y reduciendo solapes en los desarrollos.

Tipos de documentos

La organización bibliográfica, heredada de la del LDP, está basada en las diferentes clases de documentos. Estas clases se definen en parte por el tipo de documento, en parte por su tamaño y en parte por la tecnología usada para su composición.

Este es un gravísimo defecto formal que en la práctica no es tan problemático debido a que los recursos siguen manteniendo cierto orden y son bastante fáciles de manejar con la actual clasificación. Asumimos como obligación resolver en el futuro este defecto.

A diferencia del LDP aquí damos cobertura a muchas más clases de documentos y, a nuestro parecer, con más calidad de publicación. Así encontraremos las siguientes:

Guías

grandes documentos del tipo «libro» que suelen servir como manuales de referencia o de introducción. Suelen estar compuestos preferiblemente con TeX/LaTeX, DocBook o Linuxdoc-SGML. Su tamaño suele superar con holgura las 150 páginas.

Tutoriales

documentos de carácter puramente formativo.

Cómos

documentos más pequeños, compuestos con Linuxdoc-SGML o DocBook. Estos documentos suelen ser más prácticos, reducidos y de ámbito de aplicación más concreto que las guías. El término «cómo» es la adaptación del original inglés «howto» que es el nombre usado en la *Red* para denominar a los documentos que explican «how to do something».

Listas de preguntas frecuentes

recopilaciones de preguntas típicas y sus respuestas (en inglés FAQ). Suelen crearse dentro de foros de discusión con la idea de mejorar el servicio y reducir ruido y tráfico en la comunicación.

Conferencias

documentos que acompañan o ilustran conferencias impartidas alrededor de los temas cubiertos por LuCAS. Los formatos en este caso son extremadamente heterogéneos y a veces ni siquiera son libres. En cualquier caso son preferidos HTML o PostScript.

Universitarios

documentos creados en la universidad como apuntes o trabajos de investigación. Esta sección todavía no ha sido organizada y desarrollada como merece.

Estándares

estándares oficiales o industriales de aplicación en las áreas de conocimiento de interés. Una sección de especial relevancia es la dedicada a las RFC. Estos son los documentos que el *IETF* mantiene como normas de referencia de Internet.

Otros

documentos que o bien por su difícil clasificación en el sistema actual o porque no han sido publicados con tiempo suficiente no han encontrado una sección más adecuada.

La Trastienda

documentos en general obsoletos que son mantenidos como referencia histórica.

Humor

documentos humorísticos relacionados con unix y el software libre que circulan por la *Red*.

Glosario

desde que ORCA asume el mantenimiento y ampliación del antiguo glosario LuCAS se apoya y recomienda el uso y participación en el mismo con la aspiración de convertirlo en una referencia mundial.

Revistas

por ahora sólo se publica la edición en español de *Linux Focus* la revista electrónica dedicada a GNU/Linux traducida a más idiomas. En breve se iniciará la replicación de la traducción a español de *Linux Gazette*.

Licencias

publicación de las licencias libres más relevantes y, aunque no tengan la menor validez legal, de las traducciones disponibles de las mismas.

Páginas del manual

tradicionalmente en *IX nos referimos con esta expresión a la ayuda «on line» de esta clase de sistemas operativos.

Documentos replicados de otros proyectos

son tantos que es largo enumerarlos, a pesar de ser tan importantes como cualquiera; también son replicados dentro de la jerarquía LuCAS.

También son publicados en secciones propias documentos creados en proyectos afines que, si bien podrían ser englobados en la clasificación anterior, mantienen secciones propias para facilitar el mantenimiento y la búsqueda.

Cuando hablamos de tecnologías de composición queremos dejar claro que no se restringe la publicación a estos formatos sino que se completan con otros más adecuados para la consulta e impresión —como HTML, PostScript o PDF— en función de las herramientas de conversión disponibles.

Desarrollo cooperativo

Como es sabido la *Red* es un sistema caótico sin un orden particular o al menos sin una única dirección que fije su desarrollo y avance. También es conocido que en ella son muchos y frecuentes esfuerzos más o menos altruistas para la creación de recursos.

La comunidad linuxera no es ajena a este hecho y pese a la ausencia de «un gran plan» y de un órgano director jerarquizado consigue de hecho una coordinación aceptable a la hora de llevar a cabo sus proyectos.

En el caso que nos ocupa, la existencia de un lugar de referencia para la búsqueda y publicación de información permite que los activistas puedan saber de la existencia de recursos semejantes a los que tienen en mente, que puedan incorporar mejoras a los ya disponibles o que puedan encontrar lagunas de conocimiento que estén dispuestos a llenar.

Concluyendo, estamos en condiciones de asegurar que hablamos de desarrollos cooperativos con eficacia razonable y resultados demostrados.

Los medios tecnológicos principales que permiten este funcionamiento son:

- los recursos para el autor ofrecidos en LuCAS o a través de LuCAS;
- sistema de mantenimiento de código fuente (CVS);
- la lista de correo electrónico;
- la red de réplicas.

La Comunidad LuCAS

La comunidad LuCAS (*el Planeta LuCAS*) la forman (sin ningún orden concreto):

- quienes cargan con las labores de coordinación;
- los administradores de las réplicas;
- los autores;
- los revisores;
- los colaboradores que hacen apoyos puntuales dentro y fuera de la lista;
- finalmente la comunidad linuxera que hace uso de los recursos publicados.

En cuanto a los perfiles de los miembros «activos», se han ampliado desde los estudiantes y profesionales de la informática con predilección por Unix a una familia mucho más heterogénea entre la que se pueden encontrar, por ejemplo, estudiantes y profesionales de filologías, empresariales y carreras técnicas eminentemente no informáticas.

En cuanto a la nacionalidad o lugar de residencia podemos asegurar que se trata de un proyecto completamente internacional aunque centrado casi completamente en los continentes europeo y americano. Así encontraremos colaboradores muy activos en Argentina, España, Estados Unidos, Méjico o Venezuela pero también en otros países no hispano-hablantes como Francia, Noruega y Reino Unido aunque estos son más bien casos anecdóticos.

Metodología y herramientas del web LuCAS

Metodología de desarrollo

La metodología de desarrollo es la de un «prototipo evolutivo». Es decir, se recorre sucesivamente el *ciclo de desarrollo de software* para conseguir un refinamiento progresivo del proyecto y poder ofrecer servicio desde el primer momento.

Esta es la metodología de desarrollo típica de los proyectos de software libre en *Internet* que forma parte del llamado «modelo bazar».

Herramientas

Árbol ftp

LuCAS entero ha sido diseñado como un árbol ftp, sin enlaces absolutos, para facilitar su replicación vía web o ftp.

Web

- Se ha realizado un diseño simple y funcional basado en tablas;
- se han aplicado normas estándar (en la actualidad *HTML 4.0 Transitional*);
- se ha prescindido del uso de «marcos»;
- se ha facilitado el mantenimiento con la herramienta WML, guiones de shell, make y cron;
- los gráficos usados son pocos, pequeños y optimizados;
- sencilla estructura arborescente del web: un índice a la izquierda común para la mayoría de las páginas y 2 niveles de profundidad en el árbol web.

Replicación

La replicación de los recursos creados por otros proyectos para su publicación se hace mediante guiones que invocan a clientes de protocolos Internet de transferencia de ficheros —en general HTTP y FTP— lanzados mediante el reloj del sistema. La labor de estos guiones es a veces completada por código WML que genera páginas adaptadas automáticamente a los recursos replicados.

CVS

El equipo de mantenimiento accede al código fuente mediante un servidor CVS.

Correo electrónico

Se hace uso del correo electrónico para la coordinación del equipo de mantenimiento.

Cifras

Presentamos unas cifras que sirvan para ilustrar un tanto el volumen del proyecto. No deben ser tenidas en cuenta con excesiva seriedad puesto que algunas de ellas no han sido calculadas con extremo rigor científico. Cuando no han podido obtenerse datos exactos se han hecho estimaciones a la baja.

- Personas suscritas en la lista con interés manifiesto: 230.

- Número de réplicas: Argentina 2, Brasil 1, Chile 2, Colombia 2, Costa Rica 1, Cuba 1, Ecuador 1, España 12, México 5, Nicaragua 1, Perú 2 y Venezuela 1. Total 31. Además las redes LDP y Metalab publican parte de los contenidos publicados en LuCAS.
- Tamaño del árbol ftp: más de 8600 ficheros ocupando más de 320 Mb.
- Número de visitas al web (no de hits): 463000 desde el 18/2/99 hasta octubre de 2000.
- En noviembre de 1999 se hizo una estimación del valor económico de los contenidos: 2.420.592 palabras (y no incluye a todos los documentos) por 12 pts por palabra (coste de traducción) hacen un total de 29.047.104 pts (US\$181.544). No se incluyen gastos como los de publicación, conexión de red y demás mantenimientos.

Licencias de documentación abierta

Al igual que con el software, se han creado una serie de licencias orientadas a regir la distribución de documentación «abierta», no siempre libre.

Enumeramos algunas que existen actualmente y se aplican a los documentos en LuCAS.

- Free Documentation License (FDL). Es una licencia creada en GNU y basada en la GPL, por lo que su propósito es garantizar la libertad del documento. Introduce el concepto de texto transparente (frente a opaco), equivalente al código fuente en el software. Hace hincapié en la forma de construir un documento libre según este concepto.
- OpenContent License. Elimina la libertad del precio (no permite vender documentos en formato electrónico por red; mientras que sí lo permite en concepto de cubrir gastos del medio, como el CD-ROM). Acepta además la inclusión de secciones de contenido cerrado (no sujeto a esta licencia) e incide en la ausencia de garantías.
- Open Publication. Permite introducir la prohibición de realizar versiones «conceptualmente modificadas» (es decir, que cambien el significado del documento y no meramente su formato, por ejemplo) y la de comercializar versiones impresas del documento.

Como puede verse, algunas de estas licencias se diseñan con el fin de reducir el miedo editorial a trabajar con documentación abierta, eliminando para ello parte de su libertad.

Desarrollos futuros

En el marco de LuCAS se ha desarrollado una sensibilidad hacia las tecnologías documentales y bibliográficas que están sirviendo de caldo de cultivo para una serie de proyectos que servirán de base para bibliotecas electrónicas del futuro:

- Tecnologías y herramientas para la creación y mantenimiento de documentos estructurados con formatos de marcas (SGML/XML).
- Desarrollo avanzado de glosarios, diccionarios correctores y tesauros.

- Creación de un libro de estilo para la creación de documentos informáticos.
- Desarrollo de herramientas de control de las asignaciones y control de versiones.
- Diseño de un sistema bibliográfico para la publicación distribuida de documentos electrónicos.

Glosario

DocBook

Una completa DTD de aplicación informática <http://www.docbook.org>¹. Ver SGML.

DTD

Definición de tipo de documento. Ver SGML.

Fidonet

Red de correo electrónico creada por aficionados y administradores de BBS de todo el mundo. <http://www.fidonet.org>².

FSF

Ver GNU.

GNU

<http://www.gnu.org>³, <http://www.fsf.or>⁴.

GNU/Linux

Nombre por el que se conoce al sistema operativo formado por el conjunto de utilidades de sistema GNU y el núcleo Linux más otras aplicaciones libres creadas por terceros.

GPL

<http://LuCAS.HispaLinux.ES/htmls/gpl.html>⁵. Ver GNU.

LDP

Proyecto de documentación de Linux <http://www.linuxdoc.org>⁶.

LDP-es

Proyecto de documentación de Linux. Encontrará buena parte de sus contenidos publicados en <http://LuCAS.HispaLinux.ES>⁷.

Linuxdoc-SGML

Sencilla DTD de aplicación informática. <http://www.linuxdoc.or>⁸. Ver SGML.

LuCAS

<http://LuCAS.HispaLinux.E>⁹.

iBiblio.org

<http://ibiblio.org>¹⁰.

Réplica

copia completa del árbol ftp mostrada a través de un servidor web o ftp.

RFC

<ftp://sunsite.rediris.es/docs/rfc>¹¹.

SGML

Standard Generalised Markup Language (lenguaje generalizado estándar de marcas) <http://www.oasis-open.org>¹².

XML

<http://www.w3.org/XML/>¹³, <http://www.xml.org/>¹⁴. Ver SGML.

Bibliografía

Además podrá encontrar material actualizado de este tutorial y otros relacionados en

- <http://slug.hispalinux.es/~olea>¹⁵
- <http://lucas.hispalinux.es/htmls/presentaciones.html>¹⁶

Notas

1. <http://www.docbook.org>
2. <http://www.fidonet.org>
3. <http://www.gnu.org>
4. <http://www.fsf.org>
5. <http://LuCAS.HispaLinux.ES/htmls/gpl.html>
6. <http://www.linuxdoc.org>
7. <http://LuCAS.HispaLinux.ES>
8. <http://www.linuxdoc.org>
9. <http://LuCAS.HispaLinux.ES>
10. <http://ibiblio.org>
11. <ftp://sunsite.rediris.es/docs/rfc/>
12. <http://www.oasis-open.org>
13. <http://www.w3.org/XML/>
14. <http://www.xml.org/>
15. <http://slug.hispalinux.es/~olea>
16. <http://lucas.hispalinux.es/htmls/presentaciones.html>

LuCAS: un proyecto de documentación libre en español

Creación en LuCAS: Ejemplos

Juan José Amor

David Escorial

Ismael Olea

Palabras clave de este documento son: LuCAS, GNU/Linux, LDP, LDP-es, documentación, libre, traducción, revisión, creación.

Tabla de contenidos

Introducción.....	29
Proyectos de traducción o revisión.....	29
Integración de proyecto externo.....	31
Cómo conseguir el éxito y la fama.....	33

Introducción

En este documento expondremos a modo de ejemplo el ciclo de desarrollo que se sigue cuando se plantea la puesta en marcha de un proyecto en LuCAS.

Distinguiremos en LuCAS dos grandes grupos de proyectos:

- aquellos cuyo trabajo es realizado íntegramente utilizando los recursos de LuCAS (traducción o revisión puesta en marcha por un miembro de LuCAS)
- y aquellos otros que, habiéndose realizado externamente a LuCAS, desean hacer uso de algunos recursos de LuCAS para facilitar su difusión.

El trabajo en LuCAS no se reduce básicamente a estos dos casos, sin embargo son dos ejemplos que cubren un espectro muy amplio del trabajo total realizado.

Proyectos de traducción o revisión

Los roles

En un proyecto de traducción o revisión, aparecen en LuCAS dos grupos de roles.

1. *El líder*. Es la persona encargada de coordinar todo el proceso. El líder debería conocer:

- Las herramientas de autor utilizadas en el documento (LaTeX, Docbook).
- la estructura completa de archivos del documento.
- los recursos LuCAS, por ejemplo el CVS, ya que en este caso será el responsable absoluto del árbol CVS de ese proyecto.

Además de lo anterior, el líder debe tener ciertos dotes de *liderazgo*, de ahí su nombre. Es decir, capacidad para controlar a los colaboradores y también para motivarlos.

2. *Traductores o revisores*. Son las personas que, a las órdenes del líder, van realizando el trabajo asignado por éste. También deben tener conocimientos técnicos similares a los del líder, pero en menor medida. Así será deseable que el colaborador conozca:

- la herramienta de autor usada, pero este conocimiento puede limitarse a distinguir el texto a traducir de cualquier *etiqueta* o atributo del mismo. También,
- manejo básico CVS: subir cambios, actualizar y resolver conflictos.

Es muy importante hacer hincapié en que *no es necesario ser informático* para participar en un proyecto de esta clase. En general, basta con un poco de sentido común para manejar los ficheros a traducir o revisar, tener conocimientos suficientes del inglés técnico y saber escribir correctamente en español.

El ciclo de desarrollo

Las fases que se siguen con la puesta en marcha de un proyecto de esta clase serían:

1. *Planteamiento* de la idea: el futuro coordinador se plantea la idea de traducir o crear un nuevo documento.
2. *Presentación* de la idea al resto de los miembros de LuCAS, con el fin de obtener apoyos y colaboraciones necesarios.
3. Se establece un *plan de trabajo*, es decir, se divide el trabajo a realizar en las secciones que se asignarán a cada colaborador. Este plan se realiza teniendo en cuenta, el plazo para el que se quiere terminar el trabajo, los colaboradores disponibles y la cantidad de trabajo diario asumible por cada colaborador (páginas diarias).
4. A lo largo del proceso, hay una *comunicación* entre todos los miembros del proyecto, mediante la lista de correo, con el fin de discutir aspectos terminológicos, técnicos o cualquier otra duda que surja. También mediante este medio, el coordinador comunicará a todos los miembros el estado del desarrollo (capítulos entregados, capítulos pendientes, capítulos fuera de fecha que deben ser re-signados...).
5. Tan pronto como sea posible, y cuando el coordinador lo considere conveniente, se montará y *publicará una versión alfa* del documento, con todo lo realizado hasta el momento. Esta versión se publicará y servirá también para que la comunidad de usuarios pueda conocer la evolución del proyecto.
6. Cuando los miembros del proyecto tengan todos los capítulos traducidos, o en un alto porcentaje, el coordinador puede decidir crear el *equipo de revisión* del documento.
7. Por *refinamiento progresivo* se van lanzando nuevas versiones del documento, mejor traducidas y con menos errores. Esto correspondería casi a la fase de *mantenimiento* del ciclo de vida, y por tanto no tiene un fin previsto: el coordinador puede decidir en cualquier momento el lanzamiento de una nueva versión.

Esta enumeración de fases es orientativa, y en cada caso particular puede variar poco o bastante. Por ejemplo, hay coordinadores que no publican ninguna versión del documento hasta estar totalmente traducido, y en otros casos se prefiere liberar la primera versión *alfa* cuando hay un porcentaje traducido suficiente (por ejemplo, del 60 por ciento).

Facilitando el ciclo de vida con CVS

Mediante CVS el coordinador del proyecto puede facilitar la entrega de capítulos por parte de los colaboradores, e incluso facilitar que éstos puedan tener en sus sistemas copias completas del documento parcialmente traducido, de manera que puedan ellos mismos revisar su trabajo generando formatos de salida mediante la herramienta utilizada (LaTeX, etc).

- El coordinador deberá *crear una base de trabajo* con todos los ficheros que forman parte del paquete a traducir (incluyendo también aquellos que no se traducen, como los *makefiles*).
- Tras la asignación de trabajos, cada participante *accederá por CVS a los ficheros* del proyecto, los irá modificando y enviando las versiones nuevas al CVS.

- Periódicamente, o en el momento de generar una versión *alfa* del proyecto, el coordinador extraerá todos los ficheros de la base de trabajo y hará el trabajo de integración y generación, es decir, corregirá errores de sintaxis de la herramienta usada (LaTeX, etc) para conseguir generar versiones de salida del documento, que publicará. Al tiempo, el coordinador enviará al CVS los ficheros que haya modificado para que los colaboradores puedan seguir trabajando con los ficheros ya corregidos.

Liderando el proyecto

El líder debe intentar aplicar sus capacidades de liderazgo para conseguir, en la medida de lo posible, cumplir las fechas de liberación de versiones. Para ello, podemos dar algunas indicaciones orientativas:

- *Consultar a los colaboradores* cuántas páginas diarias están dispuestos a traducir o revisar. Debe exigírseles a los colaboradores un suficiente compromiso para realizarlo: de nada sirve que nos digan que pueden realizar 15 páginas diarias si luego entregan 10 al mes.
- *Repartir el trabajo* con los datos anteriores. Si tenemos suerte de que haya colaboradores sin trabajo asignado, dejarlos en una lista de reserva. Crear con todo esto una tabla de colaboradores, trabajo asignado y estado del mismo. Un trabajo asignado puede tener, por ejemplo, los estados:
 - Asignado,
 - Trabajando,
 - Entregado,
 - Reasignado, etc.
- Cuando se aproxime la fecha de entrega de un trabajo, contactar con el colaborador, para animarle a cumplir la fecha y preguntarle el estado del trabajo. La *realimentación* del colaborador es muy importante.
- Finalizado el plazo de entrega, dar un margen con los correspondientes avisos al colaborador que no ha cumplido las fechas. Si éste no confirma su continuidad o indica que no puede continuar, *reasignar el trabajo* a otro colaborador de los que ha finalizado o bien de la lista de reserva.
- *Comunicar con frecuencia el estado del proyecto* (tabla de capítulos y estado) a todos los colaboradores.

El objetivo es obtener un trabajo de calidad suficiente dentro de los plazos previstos. Los retrasos son casi siempre inevitables, pero se pueden reducir en gran medida con un trabajo adecuado.

En LuCAS, como en tantos otros sitios, es la capacidad motivadora y de liderazgo del coordinador la responsable última del éxito de un proyecto.

Integración de proyecto externo

El rol de coordinador

Cuando un proyecto no desarrollado en LuCAS quiere sacar provecho de los recursos de este último para su difusión o mejora, el único rol que nos interesa es el del responsable de la integración, que normalmente será la misma persona que coordinó el proyecto original.

El coordinador deberá conocer técnicamente, aparte de su propio proyecto y herramientas asociadas, las utilizadas en LuCAS:

Usará CVS para enviar sus modificaciones a la base de trabajo general de LuCAS. Estas modificaciones afectarán a guiones de shell que realizan réplicas automáticas, a páginas escritas en WML (Website Meta Language) y probablemente algunos otros como los *makefiles*.

La idea es que el coordinador de la integración trabaje de manera totalmente autónoma sobre el árbol de ficheros de LuCAS, sin necesitar casi en ningún momento la intervención de ningún coordinador general de LuCAS.

En resumen, el trabajo consistirá en,

- programar una réplica de su trabajo contenido en otro servidor
- y crear la página WML del proyecto.

Replicación técnica

El coordinador deberá programar una réplica automática en LuCAS de sus ficheros contenidos en su servidor.

La réplica irá contenida en el primer nivel del árbol de ficheros de LuCAS, es decir, si su proyecto se llama MiProyecto, será accesible la réplica desde la URL <http://lucas.hispalinux.es/MIPROYecto>

Para ello el coordinador modificará, por ejemplo, el guión de shell `replica-otros.sh`, disponible en el directorio `cron` del CVS de LuCAS, utilizando para replicar la aplicación `wget`, `rsync`, `mirror.pl` o incluso el propio `cvs`.

El guión es ejecutado de manera automática todos los días, de manera que el coordinador deberá ver reflejado el efecto al día siguiente.

Página de apoyo

- Debe crearse una página WML para facilitar a los usuarios de las páginas de LuCAS la navegación por la réplica.
- A veces es adecuado empotrar algo de código PERL en la página WML, de manera que la página HTML final se generará a partir del contenido del directorio replicado reduciendo el trabajo de mantenimiento.
- También debe modificarse, en su caso, la página `Imenu.wml` para crear una entrada de menú que apunte a la nueva página.
- La página de apoyo se creará en el directorio `wml` del árbol CVS principal.

- La actualización del servidor web es hecha de manera automática una vez al día, de manera que el coordinador puede observar el efecto esperando 24 horas.
- También puede montarse si lo desea una «maqueta» del web de LuCAS en su sistema, para poder probar lo que está haciendo antes de enviarlo al servidor principal.

Los coordinadores generales de LuCAS somos conscientes de que esto no es trivial, por lo que estamos siempre disponibles para ayudar a los coordinadores en sus primeros contactos con el sistema.

Cómo conseguir el éxito y la fama

- ¡Participando en los proyectos de documentación! ¡¡Hay 400 millones de lectores esperándote!!

LuCAS v3.5

Ismael Olea

Tabla de contenidos

OBJETIVOS	37
FASE 1	37
FASE 2	37
FASE 3	37

OBJETIVOS

- cambiar la estética de la web por otra más «moderna» sin sacrificar las ventajas del diseño actual;
- aumentar la producción y publicación de material;
- automatizar usando tablas ¿ascii/xml? todas las secciones del webo que aún no lo estén.

FASE 1

- finalizar Plan-activismo.txt;
 - control de calidad de las réplicas actuales;
 - especificación de requisitos del webo;
 - encargo del diseño;
 - escribir el código de la página tipo;
 - páginas mantenidas por lucas
 - páginas que replican info
 - páginas que publican material del cvs
-
- crear en una jerarquía alternativa las nuevas páginas reutilizando todo el esquema de Makefiles y todo el wml que sea posible/necesario;
 - pruebas y comprobaciones de la nueva estructura;
 - cambio al nuevo web.

FASE 2

- Herramienta de publicación: Crear una herramienta web para que puedan registrarse nuevos documentos a publicar en LuCAS.
- Asegurar que en iBiblio se publica todo el material publicado en LuCAS

FASE 3

Esta fase es realmente LuCAS v4, el sistema de nueva tecnología que está documentado en *“Arquitectura de desarrollo de proyectos de documentación libre (Diseño e implementación de LuCAS v4)”*

Web de LuCAS: Directrices de diseño

Juan Jose Amor

Tabla de contenidos

Objetivos	41
Estado actual de la web.....	41
Nuevo web: principio general.....	41
Características del nuevo web	41

Objetivos

El presente documento establece las recomendaciones a seguir para la realización del diseño para LuCAS V3. No se trata de las recomendaciones para la realización de LuCAS V4, ya que conlleva cambios en la organización de los contenidos de los que se va a prescindir por ahora.

Estado actual de la web

LuCAS V3 es un web muy valorado por los contenidos, que priman sobre el diseño estético. Una de sus características es el fácil acceso a toda la documentación que presenta al visitante. Esto se debe a la existencia de un menú desde el que permanentemente se tiene acceso a todas las páginas, así como a la abundante documentación que clasifica cada página. Todo ello acompañado de un diseño sencillo, basado en tablas y con pocos gráficos, que presenta una doble ventaja: por un lado requiere transferencia de poca información por la red. Por otro, no supone apenas carga de procesamiento sobre el navegador utilizado. Además, la separación clara de cabeceras, menús y contenidos de las páginas permite una fácil generación automática de las mismas.

Como defecto del web actual podemos citar fundamentalmente su escasa vistosidad, que pretendemos subsanar en la versión 3.5. Otros defectos, son la escasez de herramientas de consulta o búsqueda internas y una mejor organización de los contenidos. Pero esto es algo que se reserva para LuCAS V4.

Nuevo web: principio general

Como principio general, el nuevo web que se diseñe deberá como mínimo conservar las ventajas del diseño actual, y se reducirá a un cambio estético.

Características del nuevo web

LuCAS 3.5 solo será validado si cumple las siguientes condiciones:

1. La organización de los contenidos no cambiará.
2. Existirá un menú que permitirá acceder a todas las secciones de LuCAS desde cualquier sección.
3. Ninguna página deberá ver reducida la cantidad de información que puede mostrar.
4. Su lectura será fácil en los navegadores más utilizados en el mercado (Netscape Communicator 4.x e Internet Explorer 5.x).
5. La complejidad de la página seguirá siendo baja, facilitando el proceso de la misma por el navegador utilizado.
6. El diseño separará claramente cabeceras, contenidos y menús para la fácil automatización de la generación.
7. El diseño será de libre uso, respetando los derechos de reconocimiento de su autor.

Serán además deseables las características siguientes:

1. El diseño debe ser más atractivo visualmente.
2. El diseño debería elegir tipos de letra que, mejoren la visibilidad y permitan concentrar más información por página.
3. Se valorará el aumento de información en las páginas con la incorporación de *cajas de texto* o similar que permitan el acceso a diccionarios, otras herramientas de autor, otros webs de interés, etc.
4. El uso de gráficos debe ser ligero en lo posible y sin que resten accesibilidad a los textos (ver más abajo).
5. El diseño se debe acompañar de logotipos basados en el texto «Proyecto LuCAS» o similar.

A nivel técnico debe cumplirse:

1. Uso de normas HTML estándar del W3C. Se recomienda especialmente el uso de la norma XHTML. El código deberá ser validable mediante una herramienta fiable.
2. Uso de hojas de estilo en cascada (CSS), para abstraer los documentos html y facilitar el mantenimiento del web.

Y además debe tenerse en cuenta:

1. que nuestros lectores usan tanto navegadores en Windows (IE5, Netscape) como Linux (especialmente Netscape);
2. el uso de una norma estándar facilita la representación adecuada en cualquiera de estas plataformas;
3. factores de accesibilidad que facilitan el uso de navegadores poco comunes (links, lynx, nettamer) y de usuarios con minusvalías de visión;
4. que desde la coordinación tendremos en cuenta sugerencias que mejoren este documento;
5. que si bien estas indicaciones parecen muy estrictas queremos que el artista gráfico se sienta con libertad y creativo para mejorar la calidad de este proyecto al que tanto cariño le tenemos. :-)

Arquitectura de desarrollo de proyectos de documentación libre (Diseño e implementación de LuCAS v4)

Ismael Olea

Se propone una arquitectura genérica capaz de resolver todos los problemas y cuellos de botella típicos que aparecen en los proyectos de documentación libre. Además esta tecnología es apta para ser aplicada en otros proyectos de documentación. Esta tecnología será implantada con el nombre de LuCAS v4. Otras palabras clave: software libre, open source, documentos transparentes.

Tabla de contenidos

Antecedentes.....	45
Necesidades de un proyecto de documentación de soft libre	45
Técnicas que resuelven los objetivos.....	45
Herramientas que resuelven los objetivos.....	46
Sistema de publicación y consulta distribuido de nueva generación: Donantonio. 46	
Formato estructurado para los documentos.....	46
Libro de estilo.....	47
Otros recursos lingüísticos.....	48
Taller de creación y mantenimiento de documentos: TallerDoc.....	48
Generación automática de formatos de publicación: Impresor-e.....	50

Antecedentes

En base a la experiencia acumulada en años de participación y desarrollo de proyectos de documentación libre se han identificado una serie de problemas para los cuales no han aparecido soluciones completas o siquiera incompletas.

En lo que respecta a los autores su experiencia ha pasado principalmente por su participación en el proyecto LuCAS. Para conocer más sobre LuCAS les remitimos al documento *“LuCAS: un proyecto de documentación libre en español”*.

El avance más importante en LuCAS se experimentó en su v3 la cual sirvió de herramienta de investigación, experimentación y desarrollo (al estilo de los proyectos bazar) de técnicas que beneficiaran a la comunidad de usuarios de documentación libre.

Dado que v3 no ofrecía las soluciones necesarias, ni por asomo, se ha estudiado un nuevo diseño, partiendo desde cero, que ofrezca soluciones prácticas e inmediatas. Ese proyecto es llamado genéricamente LuCAS v4 y consta del desarrollo de la aplicación y su implementación en un sitio internet a fin de probarlo, mejorarlo si cabe y refinarlo.

Con todo se confía en que la comunidad internacional de desarrolladores de documentación libre haga suyo el proyecto creando una red mundial de servidores y una gran biblioteca mundial con un nivel de ruido en la información prácticamente nulo.

Necesidades de un proyecto de documentación de soft libre

- Publicación de documentos: rápidamente, sin cuellos de botella, independientemente de la zona geográfica ni del servidor.
- Formato de publicación adecuado: estándar, abierto, versátil, capaz de ser procesado automáticamente y con filtros a abundantes formatos de representación.
- Guía de estilo para la autoría y la publicación: con indicaciones para el uso del idioma, de los formatos de publicación, del sistema de publicación y del resto de recursos disponibles.
- Agilizar el trabajo del autor/traductor automatizando algunas tareas y acelerando otras.
- Entorno de trabajo en grupo:
 - foros de discusión
 - mantenimiento de código fuente
 - simplificación de la gestión de recursos humanos y del control de proyectos (asignación de tareas y control del estado de los documentos)
- simplificar el uso de formatos de publicación.

Técnicas que resuelven los objetivos

- Sistema de publicación y consulta distribuido de nueva generación.
- Ofrecer un formato estructurado para los documentos

- Libro de estilo.
- Asistentes para la autoría/traducción.
- Taller de creación y mantenimiento de documentos:
 - servidor de listas de correo;
 - mantenimiento de código fuente;
 - asistente de coordinación:
- Impresión electrónica: generación automática de formatos de publicación a partir de los formatos elegidos.

Herramientas que resuelven los objetivos

- donantonio
- docbook, linuxdoc-sgml y html
- libro de estilo
- otros recursos lingüísticos
- TallerDoc:
 - servidor mailman;
 - cvs;
 - ver más abajo.
- Impresor-e: db2* sgmltools 1.x y html2 + convenios de formato/presentación/adaptación al formato de código fuente y utilidades y automatismos.

Sistema de publicación y consulta distribuido de nueva generación: Donantonio.

Se habla sobre Donantonio en *“Introducción a Donantonio”*.

Formato estructurado para los documentos

La elección es clara. Es preciso optar por formatos completamente estructurados basados en las tecnologías SGML/XML.

Se presentan por riguroso orden de preferencia.

docbook

Hoy día es el formato estándar de documentación técnica libre. Cubre todas las necesidades de la mayor parte de los casos de documentación informática, técnica y puede ser usada perfectamente en otros muchos casos.

Está disponible en versiones SGML y XML.

Sufre carencias como la de no poder trabajar con fórmulas pero esto es algo que en el futuro podría solucionarse mediante el uso de tecnologías XML como «namespaces» y «schemas».

Al ser una herramienta desarrollada desde hace varios años ya están disponibles filtros y sistemas de impresión libres y comerciales con formatos de salida populares.

linuxdoc-sgml

Este formato ha sido muy importante porque supuso el primer paso en la estructuración de la documentación libre. Sin embargo sufre de muchas carencias en el etiquetado que docbook soluciona.

Existen algunos filtros para transformar desde linuxdoc-sgml a docbook, pero si no se mejoran precisan de asistencia y revisión.

A falta de pan buenas son las tortas pero en cualquier otro caso recomendamos enca-ricidamente usar docbook.

html

HTML es de todos estos formatos el más conocido con diferencia. Sin embargo sus pobres características para estructurar documentos lo hacen completamente desaconsejable excepto para casos de emergencia, y aún así también debería estar prohibido.

Su uso como formato de representación no entra dentro de esta sección (que está dedicada al código fuente) sino en otra parte del documento.

Sólo se podría admitir su uso para documentos extremadamente sencillos y siempre que se use la norma estrictamente, pero sigue siendo igualmente desaconsejable al estar disponible la tecnología DocBook.

Libro de estilo

NOTA: Como David Escorial tiene las ideas muy claras sobre este tema le delegamos toda la responsabilidad en convertir la sección en algo decente, interesante y útil para una sociedad capaz de engendrar a tipos como él.

Redacción de un manual completamente nuevo o recomendación de uno ya existente más ampliaciones específicas.

El libro de estilo es una herramienta absolutamente indispensable para todos aquellos que sin ser profesionales de la documentación necesitan crear textos destinados al público. Así pues el libro de estilo es la guía y oráculo que debe ayudar a resolver todas las dudas e inquietudes del autor.

Así debería contemplar al menos las siguientes secciones:

Convenios de escritura

donde se proponga una norma de estilo literario dentro del documento o de la colección de los mismos.

Vocabulario

donde se proponga una terminología uniforme a fin de no despistar al lector.

Glosario de términos técnicos

algo que parece imprescindible en el caso de la documentación informática en el que tanta importancia tiene el vocabulario de la jerga en inglés. Es obvio que debe estar perfectamente sincronizado con el vocabulario o que ambos formen una unidad.

Convenios de uso de marcas

con consejos y recomendaciones del uso de cada tecnología de marcas recomendada, concretamente de DocBook.

Convenios tipográficos

que reflejen la norma y convenios del idioma usado e instruyan su aplicación en la tecnología de documentos elegida.

Recursos de ayuda al autor/traductor

bibliografía electrónica e impresa de interés.

Técnicas de revisión

que sugiera la manera de realizar las revisiones y ofrezca indicaciones de cómo llevarlas a cabo.

Otros recursos lingüísticos

En esta sección es el «cajón de sastre» del proyecto. Cualquier herramienta cuyo uso sea recomendable y no encaje en cualquier otra sección del documento tiene aquí su sitio.

Pueden considerarse, por ejemplo:

- correctores ortográficos;
 - herramientas de traducción;
 - editores de texto optimizados;
 - correctores gramáticos;
 - diccionarios, glosarios;
 - etc.
-
- Asistentes para la creación de DTD.
 - Hojas de estilo para la transformación a otros DTD y a formatos de visualización.
 - Tesoros.
 - Asistente para la creación de meta-datos.

Taller de creación y mantenimiento de documentos: TallerDoc.

Servidor de listas de correo: mailman

Buen gestor que mantiene automáticamente un archivo web de la actividad del foro.

Mantenimiento de código fuente: cvs

Una herramienta que permite que varios autores trabajen sobre un mismo documento sin perjuicios en el trabajo de ninguno.

Existen interfaces texto y gráficos para las plataformas más extendidas.

Además conviene disponer de alguna de las herramientas de acceso a repositorios cvs vía web como cvsweb.

Asistente de coordinación:

La vida de un documento-e pasa básicamente por estas fases:

- creación
- revisión
- publicación
- lectura

Conviene mantener un control del estado del documento a fin de poder garantizar o al menos conocer la calidad de un documento.

También debe facilitarse el trabajo cooperativo y el reparto de cargas de trabajo de autoría, traducción y/o revisión, todo lo cual está facilitado por el uso de Internet pero cuya coordinación exige una gran cantidad de tiempo si no se mecaniza.

Además es razonable pensar que a pesar del proceso de calidad pueden escaparse erratas y que, dada la caducidad de la información técnica en informática, documentos completos o partes de ellos pueden quedar obsoletos con el tiempo.

Por tanto parece razonable tener en cuenta otra necesidad:

- detección y corrección de erratas.

Las erratas y su solución pueden aparecer y resolverse en cualquier momento. Si el sistema de gestión de documentación está bien integrado la publicación del documento actualizado podría ser inmediata y automática.

Para dar solución a estas necesidades se contemplan las siguientes alternativas:

- aplicación ad-hoc basada en web+sgbd;
- gestor de partes (p.e. debugs, bugzilla);
- aplicación ad-hoc basada, por ejemplo, en Narval o en la herramienta WISE.

La herramienta resultante debe contar al menos con un interfaz web.

Generación automática de formatos de publicación: Impresor-e

El Impresor-e es la herramienta que debe generar, de forma automática y completamente desatendida, los formatos de publicación (texto opaco según la terminología de la GFDL).

Adicionalmente pueden considerarse otros formatos menos estructurados pero que por tradición y por disponibilidad de herramientas hacen viable la implementación de los filtros automáticos en Impresor-e.

Se considera interesante partir de alguna clase de sistema de generación de ficheros «Makefile» adaptados al formato de código fuente.

Así mismo parece probable la necesidad de establecer alguna clase de convenios que simplifiquen el uso de la herramienta en cuyo caso deberán estar obligatoriamente documentados en el Libro de estilo.

docbook

linuxdoc-sgml

html

texinfo

LaTeX

Sólo si se usa de forma limpia y admisible por los filtros de generación de los formatos de publicación.

man

probablemente macros mm de troff :-)

Publicación de documentos libres en Internet

Ismael Olea

Tabla de contenidos

Definiciones	53
Enfoque de este artículo	53
Documento transparente y documento opaco.....	53
La publicación web y la publicación de documentos.....	53
Solución al problema de la publicación web de documentos	53
Cómo debe publicarse un documento-e (libro-e) «LIBRE»	53

Algunas ideas sobre cómo deben publicarse documentos libres en internet.

Este doc debería integrarse perfectamente con “*Arquitectura de desarrollo de proyectos de documentación libre (Diseño e implementación de LuCAS v4)*” e incluso desaparecer.

Definiciones

Definición de documento

Convendría al menos incluir la definición del DRAE y tal vez de algunas enciclopedias importantes como la Brittanica, la Espasa, etc.

Definición de documento estático (libro-e)

TACA

Definición de documento dinámico

TACA

Definición de «libre» y de «libertad» de acceso a la información

TACA

Enfoque de este artículo

Este documento se centra en la publicación de documentos estáticos libres

Documento transparente y documento opaco

Uso y explicación de las definiciones que RMS hace en la FDL.

La publicación web y la publicación de documentos

un web no suele ser necesariamente equivalente a un documento

el problema está en que se han confundido las tecnologías de acceso a la información (infraestructura de acceso) con las tecnologías de publicación (infraestructura de publicación); en concreto se confundido el uso de HTML tanto para una infraestructura como para la otra.

Solución al problema de la publicación web de documentos

suponiendo que realmente no se pueda prescindir de HTML (e incluso de XML) como infraestructura de acceso a la información es necesario separar funcionalmente (¿es esta la mejor palabra?) el documento de la infraestructura de acceso.

Cómo debe publicarse un documento-e (libro-e) «LIBRE»

- documento transparente
- código fuente estructurado
- formatos de publicación:
 - versiones navegables (html)
 - versiones imprimibles (ps, pdf)
 - otras versiones (rtf, word)
 - versiones de los anteriores empaquetados (tar/gz, zip)
- incorporación de meta-información (rdf, OMF, Dublin Core)
- publicación a través de «bibliotecas electrónicas»

Documentos estructurados

Ismael Olea

Introducción, referencias y tutoriales a formatos estructurados de documentación.

Tabla de contenidos

Introducción.....	57
Ventajas de la estructuración de documentos	57
Herramientas que asistan a la creación	57
Referencias sobre docbook	57

Introducción

Esta introducción es sólo un borrador, habría que darle un poco más de cuerpo.

Resumen de las ideas a desarrollar en la introducción (sin solaparse con la sección de nombre *Formato estructurado para los documentos en "Arquitectura de desarrollo de proyectos de documentación libre (Diseño e implementación de LuCAS v4)"*):

- la sección de nombre *Ventajas de la estructuración de documentos*
- ventajas de sgml/xml
- elección de formatos
- el futuro es XML.

Ventajas de la estructuración de documentos

- Marcación descriptiva versus marcación procedimental.
- El autor se centra en la labor de escritura y creación, sin distraerse con asuntos que no son relevantes en el proceso de autoría.
- El documento es independiente de la plataforma, facilitando su manejo, almacenado, consulta y proceso.
- Sencillez de edición y fácil manejo con cualquier clase de herramientas de edición de textos.
- Apoyado por un adecuado sistema de representación mejora la calidad final de los documentos y unifica el uso de las convenciones de estilo de cada idioma y área de conocimiento.
- Gracias al riguroso diseño también se facilita la automatización del proceso y el intercambio de documentos por medios informáticos.

Una introducción a las tecnologías de documentos estructurados está disponible en la conferencia SGML para autores¹.

Herramientas que asistan a la creación

- Biblioteca de DTD, definiciones de tipos de documentos.
- Asistentes para la creación de DTD.
- Hojas de estilo para la transformación a otros DTD y a formatos de visualización.
- Editores estructurados.
- Editores de texto.
- Asistente para la creación de meta-datos.
- Tecnologías y herramientas de representación electrónicas y en papel.

Referencias sobre docbook

Norma. Web de Norman Walsh. Libro oficial. Hojas de estilo y herramientas. Tutoriales. Tutorial en español de Ismael si es que alguna vez lo va a escribir el muy capullo. Otros DTD basados en docbook (website y slides)

Notas

1. <http://lucas.hispalinux.es/Presentaciones/campus-party98/conf-olea-cp98.tgz>

Recursos lingüísticos: herramientas lingüísticas para el autor

Ismael Olea

Tabla de contenidos

Introducción.....	61
-------------------	----

Introducción

Herramientas y convenios para facilitar el trabajo de autores y traductores.

- libro de estilo

El libro de estilo es una herramienta absolutamente indispensable para todos aquellos que sin ser profesionales de la documentación necesitan crear textos destinados al público. Así pues el libro de estilo es la guía y oráculo que debe ayudar a resolver todas las dudas e inquietudes del autor.

Cuidarse de que esto esté en sincronía con la sección de nombre *Libro de estilo* en “*Arquitectura de desarrollo de proyectos de documentación libre (Diseño e implementación de LuCAS v4)*”.

- glosario (aclarar las diferencias entre glosario, diccionario y tesoro y definir las necesidades de cada uno).
- Corrector ortográfico.

El asistente de coordinación de LuCAS v4

Ismael Olea

Tabla de contenidos

Introducción.....	65
-------------------	----

Esta parte está en pleno proceso de definición de requisitos y de implementación de prototipos.

También necesita de asegurarse que está en sincronía con *“Arquitectura de desarrollo de proyectos de documentación libre (Diseño e implementación de LuCAS v4)”*.

Introducción

Para poner en marcha la herramienta que de solución a las necesidades expresadas en la sección de nombre *Asistente de coordinación*: en *“Arquitectura de desarrollo de proyectos de documentación libre (Diseño e implementación de LuCAS v4)”* consideramos basarnos en Bugzilla¹ con una configuración adaptada a las necesidades de los proyectos de documentación y unos scripts que faciliten el trabajo.

En principio tenemos en cuenta las siguientes tareas:

- el coordinata da de alta al proyecto;
- el coordinata crea las asignaciones de cada proyecto (cada parte un fichero, por ejemplo);
- los colaboradores se apuntan vía web;
- los colaboradores pueden asumir vía web los trabajos pendientes;
- el coordinata puede asignar vía web trabajos o reasignarlos a placer;
- bugzilla manda avisos por correo electrónico de las asignaciones y de las cosas pendientes;
- bugzilla ofrece informes precisos y es cómodo y fácil de usar;

Las labores para poner en marcha la herramienta serían las siguientes:

- poner en marcha un bugzilla;
- diseñar la metodología de funcionamiento: estados y flujos de trabajo e implantarla;
- documentar las diferentes partes y usos del sistema.
 - introducción a la gestión de partes;
 - introducción a Bugzilla;
 - proceso de instalación y configuración de Bugzilla;
 - configuración de la metodología y estados del desarrollo de la documentación;
 - uso del sistema para usuarios (autores, traductores);
 - uso del sistema para editores/coordinadores.

Tengo fe ciega en que se apunte gente para construir el prototipo O:-)

Notas

1. <http://www.bugzilla.org>

Descripción de TallerDoc

Ismael Olea

Tabla de contenidos

Introducción.....	69
Sistema de escritura cooperativa.....	69

Introducción

TallerDoc es el conjunto de herramientas que vertebran el entorno de trabajo de los autores de documentación libre.

Se da por supuesta la conexión a una red telemática como por ejemplo Internet.

Se consideran imprescindibles:

- servidor de listas de correo, mailman
- cvs¹, ver la sección de nombre *Sistema de escritura cooperativa*
- digo yo que faltarán cosas... ¿no?

Sistema de escritura cooperativa

Las labores que debe favorecer este sistema son:

- Creación de documentos.
- Control de versiones.
- Creación cooperativa/distribuida.
- Actualización de los documentos.
- Control de calidad.

Con «responsabilidad distribuida» quiero referirme al modo en el que se desarrollan los proyectos tipo «bazar», en los cuales la responsabilidad de las tareas las adquieren voluntariamente los colaboradores en función de sus posibilidades y deseos; paralelamente el proyecto es dirigido/supervisado por un equipo más o menos reducido de líderes. Así es importante que las herramientas agilicen las tareas de coordinación al máximo liberando a los líderes del tiempo dedicado a las labores rutinarias.

Hay que desarrollar esto mucho más.

Notas

1. <http://cvs.hispalinux.es/cgi-bin/cvsweb/?cvsroot=LuCAS>

El impresor-e de LuCAS v4

Ismael Olea

Tabla de contenidos

taca	73
------------	----

taca

taca

Introducción a Donantonio

Ismael Olea

Tabla de contenidos

Introducción.....	77
Especificación de requisitos de Donantonio	77

Introducción

Donantonio es un sistema de nueva generación publicación y distribución de contenidos.

Para entender mejor qué pretende ser esta tecnología consulte *“Introducción a Donantonio”*, que si bien está obsoleto en lo que se refiere a la especificación de requisitos, sí es válido todavía como introducción.

Especificación de requisitos de Donantonio

- *“Donantonio: sistema bibliográfico de publicación distribuida automática”*
- *“Servidores Donantonio”*
- *“Clientes Donantonio”*
- *“Interfaz de usuario Donantonio”*

- *“Un día en la vida de un recurso en la red Donantonio”*
- *“Aplicaciones de donantonio”*

Introducción a Donantonio

Juan José Amor

David Escorial

Ismael Olea

Este artículo ya está obsoleto por otros posteriores pero sigue siendo de interés hasta que no se extraiga y ordene la información útil.

Donantonio es un sistema de publicación distribuido de nueva generación. En primera instancia se desarrolla para la publicación de documentos libres pero es inmediatamente generalizable para la publicación, distribución y localización de toda clase de recursos electrónicos o como sistema de catalogación genérico.

Tabla de contenidos

Requisitos.....	81
Especificaciones técnicas	81
¿Especificación funcional?	82
Sistema de consultas	83
Sistema de replicación de recursos.....	84
Esquemas bibliográficos.....	84
Consecuencias prácticas de la arquitectura Donantonio	85
Herramientas a desarrollar.....	86
Terminología Donantonio.....	86
La metáfora Donantonio.....	87
Quién es Don Antonio	87
Bibliografía	87

Requisitos

- Publicación distribuida: no importa dónde se encuentra el recurso, este puede ser localizado desde cualquier punto de la red donantonio.
- Responsabilidad de publicación distribuida: no hay jerarquías en la red de servidores; todos los servidores se tratan de igual a igual; además cualquier sistema en Internet que acepte los protocolos donantonio y que esté permanentemente conectado puede ser un servidor donantonio.
- Uso y reutilización de protocolos Internet: siempre que se sea posible, dentro de los protocolos donantonio se usarán las tecnologías internet estándar.
- Implementación libre: licencia GPL.
- Un sistema de consulta entre aplicaciones Donantonio.
- Técnicas para la recuperación de información por parte de usuarios finales.
- Facilidades de replicación automática en base a consultas Donantonio.
- Sencillez de diseño e implementación, independiente de lenguajes y plataformas a fin de que pueda ser fácilmente implementado y rápidamente difundido.

Especificaciones técnicas

El sistema Donantonio funcionará conforme a las siguientes reglas.

- Uso de descripciones bibliográficas (metadatos); cada recurso publicado necesita de un único documento que lo describa. Para ello se usarán esquemas bibliográficos estándar (bien ya existentes, bien ligeramente modificados bien de nueva factura).
- La descripción no tiene por qué estar alojada en el mismo servidor que el recurso.
- Cada servidor mantendrá sus propias descripciones, ofreciéndolas a quien esté dispuesto a copiarlas.
- Existirá una diferencia formal entre una descripción original y su copia. Esta diferencia se plasmará en una marca de originalidad-o-no. La marca de originalidad sólo la tendrá la descripción original, situada en el servidor de publicación. Cada vez que se copia una descripción original la marca de originalidad debe desaparecer.
- El hecho de publicar descripciones originales o copias se hace de forma conjunta y sin distinción.
- Las descripciones replicadas (de diferentes fuentes) pueden ser sometidas a un proceso de filtrado para eliminar duplicados idénticos. Queda por definir el algoritmo que distinga, por ejemplo, versiones diferentes, si es que es preciso (tal vez sea un filtro que deba tener disponible sólo el usuario final o el mecanismo de consulta). Lo más ideal tal vez sea disponer de «módulos de filtros» adaptables, intercambiables y modificables a placer. *NOTA: Este párrafo no se puede quedar así por nada del mundo.*
- Un servidor donantonio puede ser a su vez cliente donantonio.

- Cada cliente replicará las descripciones de los servidores que le plazcan, sin mantener ninguna topología o norma concreta fuera del libre albedrío del administrador del servicio donantonio.
- por ahora no se decide cuál debe ser el método de almacenamiento interno. Probablemente no haya que hacerlo nunca.
- Es obligatorio importar/exportar descripciones en el formato definido por la norma donantonio.
- Sistema de consulta entre aplicaciones.
- Técnicas de usuario final para recuperar información:
 - búsqueda de patrones en descriptores;
 - navegación por taxonomías dinámicas;
 - ontología bibliográfica (váyase usted a saber qué es eso).
- Procesos de replicación/filtro/consulta (POR DEFINIR).

¿Especificación funcional?

Acciones que lleva a cabo un servidor Donantonio:

- registrar descripciones
- publicar almacén de descripciones visible a través de
 - mecanismo de consulta
 - protocolos internet de transferencia de ficheros
- (opcional) publicar descripciones (o sea, publicar la dirección a través de una url como un fichero convencional más)
- atender a consultas Donantonio

Acciones que lleva a cabo un cliente Donantonio:

- replicar almacenes de descripciones mediante
 - protocolos internet de transferencia de ficheros
 - consultas Donantonio
- replicar almacenes de recursos en función de una consulta (con la que se genera previamente el almacén de descripciones).

Acciones del interfaz de usuario Donantonio

(el interfaz de usuario es una herramienta interactiva de consultas)

- Diseño de consultas (¿a estilo QBE?)
- Representación del resultado de las mismas
- Acceso a los recursos referenciados en las consultas

Sistema de consultas

Las consultas son el mecanismo mediante el cual el usuario final (humano o cibernético) accede al conocimiento y a los recursos.

Se proponen tres tipos de consulta:

(Nota: ahora que lo pienso puede esta división sólo tenga sentido a nivel de interfaz de usuario y que puedan codificarse internamente con el mismo mecanismo... ¿tal vez SQL? Es probable)

- búsqueda de patrones en descriptores bibliográficos;
- navegación taxonómica (qué nombre tan grandilocuente, espero no haber metido la gamba).
- ontología bibliográfica (chúpate esa, grep).

Búsqueda de patrones

La búsqueda de patrones es una búsqueda clásica. Si es necesario intentamos hacer una descripción más formal. El interfaz de usuario tal vez podría ser tipo QBE.

Navegación taxonómica

La navegación taxonómica creo que es una herramienta potentísima. No he comprobado si sitios como Yahoo lo usan pero es ideal.

¿En qué consiste? Fácil. Las descripciones bibliográficas están hechas en base a un esquema. Ese esquema tiene una sintaxis y contenidos predeterminados. Si ordenamos un conjunto de descripciones en función de un orden determinado de los campos que forman parte del esquema, obtenemos una jerarquía. Lo ideal es que el orden de la jerarquía se adapte a cada consulta en función del usuario. Así, una consulta podría empezar ordenando descripciones de mamíferos por terrestres/anfibios/marinos/o voladores y hacer que la siguiente jerarquía esté en función de otro campo como podría ser... joer, no me acuerdo cómo se clasifican los mamíferos.

La cuestión es que otra consulta puede tener un orden diferente.

En cada consulta el usuario va navegando más profundamente refinando la consulta.

Para que este refinamiento tenga más sentido, en cada incremento de la profundidad puede añadir restricciones en el sentido de campos que cumplan o no determinadas propiedades.

La consulta finalmente debe tener la posibilidad de ser guardada, a fin de reutilizarla en otro momento (como una vista al almacén de descripciones).

Como es de suponer, a lo largo de la navegación van apareciendo referencias a las descripciones que cumplen las condiciones de la búsqueda. Finalmente las descripciones nos llevarán al recurso. Como Freshmeat, pero bien hecho.

Sé que esto es un infierno de incongruencias, pero para eso estamos, para limpiar, fijar y dar esplendor.

Ontología bibliográfica

Ilumínanos, David.

Sistema de replicación de recursos

Si construimos una consulta Donantonio se supone que es porque tenemos alguna clase de interés en los recursos referidos. En ese caso es interesante poder disponer de ellos rápidamente en modo local.

Para eso nada más fácil que replicar los mismos manteniendo una copia sincronizada de cada uno de ellos.

Disponiendo de las descripciones de nuestro gusto y dado que los recursos son accesibles vía internet es casi trivial diseñar una herramienta que los replique.

Los usos son muchos y variados:

- creación de bibliotecas-e departamentales
- confección de cdrom recopilatorios
- mantenimiento de distribuciones Linux.

Esquemas bibliográficos

Los esquemas bibliográficos son conjuntos de descriptores de metadatos normalizados que se usan para clasificación bibliográfica. *NOTA*: apuesto a que tiene una definición más formal.

RDF

Las siglas RDF responden (SI NO ME EQUIVOCO) a Resource Description Framework, que es un estándar del W3C para la codificación de descripciones bibliográficas usando tecnologías XML.

Una de las ventajas del diseño de RDF es que, gracias a los *name spaces* de XML, una descripción puede estar almacenada en un documento XML de forma completamente transparente. Eso facilita enormemente el mantenimiento de ambos.

RDF está desarrollándose rápidamente en la industria del software (comercial y libre) y las nuevas versiones de los navegadores más importantes contemplan implementarlo.

Esquemas estándar

Siempre que sea posible se procurará usar esquemas ya estandarizados en RDF, XML o en otras tecnologías adaptándolos, si es preciso, al nuevo medio.

Algunos esquemas interesantes son

- OMF, para documentación libre;
- lsm, para software libre;
- rdf/rpm (no sé su nombre exacto) para catalogar paquetes de software RPM;
- UDDI, nueva norma para la catalogación y publicación de servicios B2B en internet;
- donantonio-biblio (pendiente de definir), para la publicación y difusión de servidores.

Ampliaciones Donantonio de esquemas estándar

No es el propósito de esta tecnología contaminar estándares abiertos, pero que esta tecnología funcione como se espera es preciso codificar cierta información en cada descripción.

Está pendiente de resolver si esta información puede codificarse transparentemente mediante los *name spaces* o es preciso crear variantes mínimas de los esquemas a usar.

Consecuencias prácticas de la arquitectura Donantonio

- Los recursos y las descripciones tienen puntos fijos de publicación y son accesibles a través de una URL.
- Es posible la existencia de recursos y descripciones idénticos con diferentes URL. Aunque precisamente se ha diseñado Donantonio con la idea de erradicar esta costumbre, la funcionalidad del mismo sólo empeora ligeramente si se usa.
- Las copias de recursos y descripciones (copias de los mismos) no tienen el mismo interés bibliográfico y por tanto tienen un tratamiento claramente separado.
- El almacén de descripciones de cada servidor donantonio es visible en cualquier parte de la red.
- Los servidores pueden ser genéricos (replican todo) o especializados (recopilan sólo cierto tipo de información de interés local).
- Las facilidades de replicación a través de consultas diseñadas convenientemente permiten redistribuir la carga de servidores a medida de las necesidades.
- Cualquier recurso localizado a través de donantonio es accesible vía internet.
- No se puede garantizar la localización de un recurso en la red donantonio, pero sí en su espina dorsal (subred de servidores genéricos).
- Los clientes donantonio podrían ser otros servidores, interfaces web o un programa cliente en cualquier lenguaje clásico.

- pueden diseñarse redundancias entre servidores (un servidor puede proveerse de dos servidores idénticos filtrando duplicados) que fortalecen la red contra caídas de algunos nodos.

Herramientas a desarrollar

- editor de descripciones
- creador semi/automático de descripciones
- publicador de descripciones
- replicador de descripciones
- gestor de descripciones
- motor de consultas;
- herramienta de usuario para navegación y recuperación de información;
- replicador de ficheros (sobre una consulta prefijada) (organiza una consulta/replicación-de-metadatos en una jerarquía de recursos que son copia de los originales)

Terminología Donantonio

Elementos

Recurso

fichero o documento accesible a través de una URL.

Descripción

fichero de metadatos de un recurso, accesible a través de una URL.

Repositorio

colección de recursos o de copias de recursos.

Biblioteca

colección de descripciones o de copias de descripciones.

Tipos de agentes donantonio

Servidor

nodo actuando como servidor donantonio.

Cliente

nodo actuando como cliente donantonio o cliente donantonio puro.

Representante

nodo que actúa a la vez como cliente y como servidor donantonio. *NOTA: tal vez este agente sea formalmente prescindible.*

La metáfora Donantonio

Para explicar, aunque de forma completa, lo que es Donantonio puede usarse la «metáfora astral».

La metáfora astral

Si creemos a algunos astrólogos o a ciertas religiones, el cuerpo físico de las personas está unido a un cuerpo astral: el alma. En ciertas circunstancias el alma puede viajar fuera de su correspondiente cuerpo, que duerme en algún lugar de la Tierra, permaneciendo ambos unidos gracias a una cinta plateada. Si la cinta se rompe por algún motivo, es la muerte para esa persona, cuyos cuerpo y alma quedan irremisiblemente separados para siempre.

Equivalencia entre la metáfora Donantonio y la metáfora astral

Según el esquema Donantonio, los recursos son los cuerpos, que permanecen fijos en algún sitio, dormidos, mientras que sus almas, que son los metadatos que los describen, viajan libremente por la red Donantonio, unidos ambos por una URL.

Si el enlace se pierde, el recurso desaparece. El documento ha muerto.

La idea principal es que mientras los recursos permanecen tranquilos, durmiendo en una URL de referencia, sus descripciones se mueven con libertad y son procesadas con libertad sin peligro a perder el nexo entre ambos que hacen que el mecanismo de la comunicación tenga lugar.

Quién es Don Antonio

Don Antonio es el nombre del bibliotecario donde estudié el bachillerato.

Elegí el nombre porque me pareció sonoro, expresivo y como una nota de color y alegría frente a los fríos nombres de los protocolos y aplicaciones de Internet.

Bibliografía

- XML
- DC
- RFC 2413 (DC)

- RDF
- Los de la clasificación automática.
- URL de LuCAS
- del borrador del libro de estilo de Insflug
- al glosario de LuCAS
- ldp-core.dtd: <http://casbah.org/cgi-bin/cvswweb.cgi/XML/DTDs/ldpCoreMetaML.dtd>
Creada por Kendall Clark en XML. Se supone un paso previo a la versión RDF.

DC

<ftp://sunsite.rediris.es/docs/rfc/24xx/2413>¹ <http://purl.org/DC/>²

DocBook

<http://www.docbook.org>³

DTD

TACATACA

GNU

<http://www.gnu.org>⁴ <http://www.fsf.org>⁵

Glosario de LuCAS

<http://LuCAS.HispaLinux.ES/Otros/glosario/Glosario/glosario.html>⁶

GPL

TACATACA

ISO

International Standards Organization

LDP

<http://www.linuxdoc.org>⁷

LDP-es

<http://LuCAS.HispaLinux.ES>⁸

LuCAS

<http://LuCAS.HispaLinux.ES>⁹

Metalab

<http://metalab.unc.edu>¹⁰

RDF

<http://www.w3.org/Metadata/RDF>¹¹

RFC

<ftp://sunsite.rediris.es/docs/rfc/>¹²

SQL

TACATACA

URL

Uniform Resource Locators <ftp://sunsite.rediris.es/docs/rfc/17xx/1738>¹³

SGML

TACATACA

XML

<http://www.w3.org/XML/>¹⁴

W3C

<http://w3.org/>¹⁵

Notas

1. <ftp://sunsite.rediris.es/docs/rfc/24xx/2413>
2. <http://purl.org/DC/>
3. <http://www.docbook.org>
4. <http://www.gnu.org>
5. <http://www.fsf.org>
6. <http://LuCAS.HispaLinux.ES/Otros/glosario/Glosario/glosario.html>
7. <http://www.linuxdoc.org>
8. <http://LuCAS.HispaLinux.ES>
9. <http://LuCAS.HispaLinux.ES>
10. <http://metalab.unc.edu>
11. <http://www.w3.org/Metadata/RDF>
12. <ftp://sunsite.rediris.es/docs/rfc/>
13. <ftp://sunsite.rediris.es/docs/rfc/17xx/1738>
14. <http://www.w3.org/XML/>
15. <http://w3.org/>

Sistema distribuido automático de publicación. (era "Donantonio, an universal indexer architecture (draft)")

Ismael Olea

Este documento está «de aquella manera»

Este documento está «de aquella manera»: Esto está bastante guarrete, pero lo pongo aquí porque parece que tiene cosas importantes, pero en cualquier caso hay que limpiarlo de polvos y de pajas.

The very basic ideas for an architecture for publishing and maintaining all kind e-libraries using known standards as XML, RDF, Dublin Core, etc.

Tabla de contenidos

Sistema automático de publicación.....	93
Sistema automático de publicación.....	93
Introducción: al principio era LuCAS.....	94
Requeriments for an automatic distributed publishing architecture.....	94
Architecture versus application.	94
Donantonio vocabulary	95
The Donantonio metaphor.....	96
Who's Don Antonio?	98
Change-log	98

Sistema automático de publicación

Actualmente estoy trabajando en la especificación de los requerimientos a exigir a esta clase de tecnologías. Es lo que he llamado la «arquitectura *Donantonio*».

Los planteamientos básicos de *Donantonio* son:

- Basado en soft libre, preferiblemente GPL o compatible.
- Especificaciones abiertas y perfectamente documentadas.
- Basado en verdaderos estándares abiertos como RFC, W3C e ISO.
- Sencillez, modularidad y flexibilidad.

Los objetivos del diseño son:

Los objetivos del diseño son:

- publicación automática;
- publicación topológicamente distribuida;
- responsabilidad distribuida de la publicación;
- organización de vistas –de usuario y de sistema– automáticas y programables;
- facilidades de replicación;
- facilidades de actualización automática;
- relaciones flexibles entre agentes *Donantonio*.

Elementos necesarios (entre otros):

- declaración de metadatos conforme a un esquema normalizado (descripción);
- el fichero como unidad atómica de transferencia de datos;
- protocolos de transferencia de ficheros internet;
- mecanismo de consulta sencillo y flexible (por ahora, SQL).

Sistema automático de publicación.

En la actualidad estoy diseñando el borrador de una arquitectura para el diseño y operación de esta clase de sistemas. Además participo en el diseño de una aplicación bibliográfica que asista a la publicación de recursos en *Metalab* con la vista puesta en un diseño basado en *Donantonio*.

También tengo previsto que este esfuerzo, junto a otros que al menos partirán inicialmente de la comunidad hispana, forme parte del próximo proyecto LuCAS I+D.

Por lo pronto ya están en desarrollo los siguientes trabajos:

LDP-core,

esquema bibliográfico para clasificar los documentos del LDP. TACATACA¹.

ldpCoreMetaML.dtd,

LDP-core expresado como DTD XML. ldpCoreMetaML.dtd²

Donantonio,

una arquitectura para publicación distribuida automática.

#####

Otra nota: No olvidar lo de #####An automatic distributed publishing system (with facilities of information searching/retrieval).#####

#####

Introducción: al principio era LuCAS

Una de las conclusiones más claras obtenidas de la experiencia en el mantenimiento y coordinación del proyecto LuCAS (<http://lucas.hispalinux.es>) y a lo largo de las relaciones con el resto de proyectos tipo LDP-es es que el mantenimiento de un sistema "clásico" de publicación basado en web es del todo inmanejable en cuanto el volumen de documentación alcanza cierto volumen.

A primera vista parece adecuado el tratamiento de la información mediante técnicas de bases de datos. Sin embargo este autor no cree que sea la mejor aproximación: el desarrollo actuales de las tecnologías de documentación han posible la puesta en marcha de sistemas "documentalmente más naturales". No se descarta el uso de técnicas de bases de datos en estos sistemas. Más al contrario: parece necesario el uso a nivel interno de las mismas pero siempre como un medio y no como un fin.

A lo largo de este documento se expondrá el sistema Donantonio, una serie de técnicas y convenios, basados en normas abiertas, para construir sistemas de publicación de nueva generación.

Requeriments for an automatic distributed publishing architecture

Inside my own vision of software engineering it must be accordant to the next rules:

- free software, with a GPL license if it's possible;
- open, its api's must be clearly document for all;
- based on true standards as RFC's, ISO and 3WC standards.
- modular and flexible to become the layer for distributed applications

Architecture versus application.

This paper is focused in architecture only. The idea is to design a set of conventions and rules to build a working environment to develop applied publishing tools.

For this cause I'll try not to say too much about the Metalab/LDP publishing system. Indeed, the designing work for these appliances will help to refine Donantonio to get a true multi purpose and generalized architecture.

Donantonio vocabulary

Elements

Resource

file or a document located via an URL.

Description

a RDF file meta-data, located via an URL.

Resource instance

Por definir

Description instance

por definir

Repository

collection of resources or collection of resources' instances.

Library

collection of descriptions or collection of descriptions' instances.

Agent tasks

These tasks aren't completely defined yet.

Server

software server

Client

software client

Proxy

Proxy between two servers or between a server and a client

Actions

These are the "atomic actions":

Resource download

the act of downloading a resource referred by an URL via an Internet transfer protocol.

Resource mirroring

the act of mirroring (and sincronizing) a resource referred by an URL via an Internet transfer protocol.

Description download

the act of downloading a description referred by an URL via an Internet transfer protocol.

Description mirroring

the act of mirroring (and sincronizing) a description referred by an URL via an Internet transfer protocol.

Query

an SQL ask from something acting as a client to something acting as a server.

And these are the "composed actions"

Library mirroring

TACA

Repository mirroring

TACA

Distributed query

TACA

Relations

Projection

relational representation in SQL language of a RDF schema.

Query

SQL query to a projection.

Petition

a query on a transport layer (mime+http??).

Answer

of a query on a transport layer (mime+http??). It can contains a view.

View

a set of one or more description Urls.

The Donantonio metaphor

The astral metaphor

An incomplete metaphor of the things Donantonio could support is the "astral metaphor".

If we believe to some astrologists, the physical body has an astral body (the soul). In some circumstances the soul can travel without its physical body. But soul and body are linked with a silver bow. The astral body (the soul) can travel to any place of the world and ever is linked to his body, which is still sleeping somewhere on the Earth. But if the silver bow get broken, the soul is not linked to its body anymore. It's the death for its body.

Equivalence between Donantonio metaphor and the astral metaphor

For Donantonio thinking, documents are the body, descriptions are the astral bodies of its documents and are linked to it by a URL. If the link get broken the description lost all is usefulness and the documents die forgotten in somebody hard disk.

The principal idea is that while resources are sleeping in some place, their souls are traveling along the world, being processed and shared between Donantonio appliances. Donantonio add the ubiquity faculty letting several duplicated souls for a only resource

Practical consequences

- Resources had fixed publishing points (one or many) referenced by a URL.
- A resource never travel across the net.
- A resource can be downloaded. The downloaded resource is an instance of the resource and it hasn't bibliographic relevance.
- Descriptions had a one and only fixed publishing point referenced by a URL.
- Descriptions instances can travel across the net to by archived or processed. A description instance has bibliographic relevance while they had a valid link with their master description.
- A repository can be a Donantonio repository while all its resources had its description in a Donantonio application somewhere in the net.
- A Donantonio library is a true library while all its descriptions or descriptions instances are valid (they refer to their own resources).
- It's too easy to replicate libraries only replicating its descriptions files. A way to distribute load distributing server accesses.
- The way for building indexing clusters or indexing servers trees is distributing the descriptions their self between the servers. An information petition on one of the servers (the cluster case) or in the main server (the tree case) is distributed to the others with a Donantonio query. Them, the systems acts as one only server.
- Descriptions can be grouped into views with SQL queries and explicit Urls.
- A distributed/monolithic library can be replicated with only a query (a view) into a monolithic mirror or with several well designed queries (several views) into a net of servers.

Sistema distribuido automático de publicación. (era "Donantonio, an universal indexer architecture (draft)")

- The resources referenced by a library (or many libraries) can be instanced locally (cached) with views and URL downloads.

Who's Don Antonio?

Don Antonio is the librarian of the high school were I had my bachelor degree.
I like this name because it's different and sounds funny :-9

Change-log

- 1999-09-18 v0.0, first release in English
- 2000-03-19 v0.02, revisión y fusión con el proyecto de Bibliotecas electrónicas.

Notas

1. TACATACA
2. <http://casbah.org/cgi-bin/cvsweb.cgi/XML/DTDs/ldpCoreMetaML.dtd>

Donantonio: sistema bibliográfico de publicación distribuida automática

Especificación de requisitos software

Juan José Amor

David Escorial

Ismael Olea

Tabla de contenidos

Introducción.....	101
Descripción general.....	103
Requisitos específicos	104
Apéndices	107

Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el sistema bibliográfico de publicación distribuida automática Donantonio. Todo su contenido ha sido elaborado teniendo en cuenta las necesidades observadas en la experiencia de los autores en publicación en Internet. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar IEEE *Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998*.

Propósito

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El documento va dirigido tanto al equipo de desarrollo, como a los integrantes de los proyectos de documentación libre y a la comunidad de posibles usuarios finales. Este documento será el canal de comunicación entre las partes implicadas, tomando parte en su confección miembros de cada parte. Esta especificación está sujeta a revisiones por las partes implicadas, especialmente por los potenciales usuarios, que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

Ámbito del sistema

Este sistema es requerido a raíz de la creciente complicación de las labores de publicación de documentación del software libre.

Se ha constatado la necesidad de un sistema informático que automatice las labores de publicación de documentos en Internet, de tal forma que se garantice la fácil localización de los mismos y una rápida puesta a disposición a los usuarios.

Usando normas bibliográficas puede generalizarse el uso y aplicación del sistema a cualquier tipo de información susceptible de ser clasificada.

El ámbito del sistema desarrollado llega hasta la realización de clientes y servidores de Donantonio, protocolos de comunicación entre los mismos e interfaces de usuario; prefiriéndose el diseño de un sistema que pueda ser empotrado en otras aplicaciones afines para que éstas puedan a su vez ser aplicaciones compatibles (clientes, servidores o interfaces) con Donantonio.

Definiciones, acrónimos y abreviaturas

Definiciones

Recurso

Elemento accesible a través de una URL susceptible de ser clasificado bibliográficamente.

Esquema

Describe la sintaxis de una descripción.

Descripción

Instancia de un esquema que identifica mediante metadatos, de manera inequívoca a un recurso.

Biblioteca

Colección de descripciones.

Servidor

Nodo que atiende consultas donantonio.

Cliente

Nodo que genera consultas donantonio.

Consulta

Petición de un subconjunto de descripciones expresada en un lenguaje normalizado.

Acrónimos

ERS

Documento de Especificación de Requisitos Software.

GFDL

Licencia para Documentación Libre de GNU.

GPL

Licencia para Software Libre de GNU.

RDF

Resource Description Framework, estándar del consorcio de World Wide Web (W3C) para la codificación y manejo de metadatos usando el lenguaje de marcas XML.

URL

Uniform Resource Locator, esquema utilizado para la localización de un recurso en Internet.

XML

eXtended Markup Language. Lenguaje de marcas estructuradas, estándar del W3C.

Abreviaturas

No se han definido.

Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.

OMF, GNU, GFDL, GPL, RDF, XML, URL? *Nota: En la lista anterior deben ir las referencias realmente necesarias. Es decir, una criba sobre la lista que aparece en especificaciones.sgml*

Visión general del documento

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema.

Descripción general

Perspectiva del producto

Este sistema no interactuará con ningún otro sistema software.

Funciones del sistema

Publicación de recursos

El sistema permitirá publicar recursos situados en cualquier lugar de la red Internet. Cada recurso será localizado mediante su descripción, que estará en los servidores Donantonio, y por tanto, no tendrá por qué coincidir con el lugar de publicación del recurso.

Registro de descripciones

Cada descripción tendrá un punto de publicación, desde el cual un servidor donantonio lo incorpora a su biblioteca a través del proceso de registro.

El registro será distribuido, en los sistemas que denominaremos *servidores donantonio*.

Consulta y replicación de descripciones

Los *servidores donantonio* incluirán mecanismos o protocolos para admitir la replicación total o parcial de las descripciones, que solicitarán los llamados *clientes donantonio*.

Para este fin, los servidores deberán poder atender consultas de descripciones, para lo que se definirán los correspondientes mecanismos o protocolos.

Mecanismos de usuario para búsquedas

La *interfaz de usuario donantonio* implementará mecanismos adecuados de navegación y búsqueda de la información.

Distribución

Las descripciones podrán ser replicadas a través de redes donantonio.

No se establece una jerarquía en la red de servidores, es decir, todos se tratarán entre iguales.

Características de los usuarios

El sistema realizado deberá ofrecer una interfaz de usuario intuitivo, fácil de aprender y sencillo de manejar. El sistema deberá presentar un alto grado de usabilidad. Lo deseable sería que un usuario nuevo se familiarizase con el sistema en muy poco tiempo, siendo de uso inmediato para las funciones de búsqueda de recursos.

Restricciones

El sistema será software libre (de acuerdo con la licencia GNU-GPL o similar) y deberán ser libres aquellos componentes que reutilice.

El sistema se diseñará según un modelo cliente/servidor.

El sistema basará sus comunicaciones en protocolos estándar de Internet.

Los distintos subsistemas deberán tener un diseño e implementación sencillos, independientes de la plataforma o el lenguaje de programación.

Los servidores deben ser capaces de atender consultas concurrentemente.

Suposiciones y dependencias

Suposiciones

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por el equipo de diseño atendiendo sugerencias de la comunidad. Cualquier petición de cambios en la especificación debe ser aprobada por el equipo de diseño y gestionada por el mismo.

Dependencias

El sistema Donantonio funciona autónomamente, sin necesidad de comunicarse con otros sistemas externos, por lo que no hay dependencias respecto de otros sistemas.

Requisitos específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son ESENCIALES, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de testabilidad : dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

Nota: Es conveniente hacer la «matriz de requisitos», que confronta los requisitos con el método de verificación.

Requisitos funcionales

REQ01: Registro de descripciones

El sistema debe registrar descripciones de recursos (metadatos y el correspondiente localizador URL).

REQ02: Visibilidad de las descripciones

Todo servidor donantonio implementará mecanismos de consulta que permitan seleccionar desde un cliente un conjunto de las descripciones que éste dispone.

REQ03: Transportabilidad de las descripciones

Todo servidor donantonio permitirá la exportación de un conjunto seleccionado de descripciones.

REQ04: Selección de descripciones

Todo cliente donantonio implementará mecanismos de consulta que permitan acceder en un servidor a un conjunto de las descripciones que éste dispone.

REQ05: Replicación de descripciones

Todo cliente donantonio podrá replicar un conjunto de descripciones obtenidas desde un servidor donantonio mediante un criterio de selección de los definidos en REQ04.

REQ06: Independencia entre servidores

Los servidores serán independientes para poder dar servicio autónomamente a los clientes.

REQ07: unicidad de las descripciones

Las descripciones deberán ser únicas en cada biblioteca.

Requisitos de interfaces externos

Interfaces de usuario

REQ08: Interfaz de usuario.

El sistema contará con una interfaz de usuario que permita realizar fácilmente las consultas de descripciones y acceder a los recursos referenciados por una consulta.

Interfaces hardware

No se han definido.

Interfaces software

No se han definido.

Interfaces de comunicación

REQ09: Protocolos estándares en Internet

Los servidores, clientes y aplicaciones donantonio se comunicarán entre sí mediante protocolos estándares en Internet, siempre que sea posible. Por ejemplo, para transferir ficheros (recursos o bloques de descripciones) deberán utilizarse protocolos existentes (FTP u otro conveniente).

Requisitos de rendimiento

REQ10: Tiempo de respuesta

Los tiempos de respuesta a las consultas interactivas deberán ser reducidos, considerando una red de comunicaciones eficiente.

REQ11: Concurrencia

El sistema deberá soportar atención concurrente a las consultas.

Requisitos de desarrollo

REQ12: Ciclo de vida.

El ciclo de vida elegido para desarrollar el sistema será el de prototipo evolutivo, de manera que se puedan incorporar fácilmente cambios y nuevas funciones.

Requisitos tecnológicos

REQ13: Normas bibliográficas

Los esquemas bibliográficos a usar en el sistema estarán expresados en RDF.

Atributos

REQ14: Software libre

El sistema será software libre y, por tanto, cualquier componente software que reutilice también deberá ser libre.

REQ15: Portabilidad

El sistema será diseñado portable. Deberá ser posible implantar aplicaciones donantonio en diversos lenguajes y plataformas.

REQ16: Mantenibilidad

El sistema es susceptible de ser ampliado. Por tanto deberá diseñarse fácilmente mantenible, aplicando para su desarrollo las metodologías que para ello sean precisas.

Otros requisitos

Apéndices

Licencia

Donantonio: Especificación de Requisitos Software

Copyright (C) 2000 Juan J. Amor, David Escorial e Ismael Olea.

Se otorga permiso para reproducir total o parcialmente este documento, siempre y cuando se cite la fuente. Cualquier modificación de este documento deberá ser aprobada por los autores.

Servidores Donantonio
Especificación de requisitos software

Juan José Amor

David Escorial

Ismael Olea

Tabla de contenidos

Introducción.....	111
Descripción general.....	113
Apéndices	117

Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el subsistema servidor del sistema bibliográfico de publicación distribuida automática Donantonio. Todo su contenido ha sido elaborado teniendo en cuenta las necesidades observadas en la experiencia de los autores en publicación en Internet. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar IEEE *Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998*.

Propósito

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El documento va dirigido tanto al equipo de desarrollo, como a los integrantes de los proyectos de documentación libre y a la comunidad de posibles usuarios finales. Este documento será el canal de comunicación entre las partes implicadas, tomando parte en su confección miembros de cada parte. Esta especificación está sujeta a revisiones por las partes implicadas, especialmente por los potenciales usuarios, que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

Ámbito del sistema

Este sistema es requerido a raíz de la creciente complicación de las labores de publicación de documentación del software libre.

Se ha constatado la necesidad de un sistema informático que automatice las labores de publicación de documentos en Internet, de tal forma que se garantice la fácil localización de los mismos y una rápida puesta a disposición a los usuarios.

Usando normas bibliográficas puede generalizarse el uso y aplicación del sistema a cualquier tipo de información susceptible de ser clasificada.

El ámbito del sistema desarrollado llega hasta la realización de los servidores de Donantonio; prefiriéndose el diseño de un sistema que pueda ser empotrado en otras aplicaciones afines para que éstas puedan a su vez ser aplicaciones compatibles con otros módulos Donantonio.

Definiciones, acrónimos y abreviaturas

Definiciones

Recurso

Elemento accesible a través de una URL susceptible de ser clasificado bibliográficamente.

Esquema

Describe la sintaxis de una descripción.

Descripción

Instancia de un esquema que identifica mediante metadatos, de manera inequívoca a un recurso.

Biblioteca

Colección de descripciones.

Servidor

Nodo que atiende consultas donantonio.

Cliente

Nodo que genera consultas donantonio.

Consulta

Petición de un subconjunto de descripciones expresada en un lenguaje normalizado.

Acrónimos

ERS

Documento de Especificación de Requisitos Software.

GFDL

Licencia para Documentación Libre de GNU.

GPL

Licencia para Software Libre de GNU.

RDF

Resource Description Framework, estándar del consorcio de World Wide Web (W3C) para la codificación y manejo de metadatos usando el lenguaje de marcas XML.

URL

Uniform Resource Locator, esquema utilizado para la localización de un recurso en Internet.

XML

eXtended Markup Language. Lenguaje de marcas estructuradas, estándar del W3C.

Abreviaturas

No se han definido.

Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.
- Donantonio: sistema bibliográfico de publicación distribuida automática. Especificación de requisitos software. J.J. Amor, D. Escorial e I. Olea, 2000.
- Clientes Donantonio. Especificación de requisitos software. J.J. Amor, D. Escorial e I. Olea, 2000.

Visión general del documento

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema.

Descripción general

Perspectiva del producto

El servidor donantonio interactuará con clientes, mediante un protocolo que deberá definirse.

Funciones del sistema

Publicación de recursos

El sistema permitirá publicar recursos situados en cualquier lugar de la red Internet. Cada recurso será localizado mediante su descripción, que estará en los servidores Donantonio, y por tanto, no tendrá por qué coincidir con el lugar de publicación del recurso.

Registro de descripciones

Cada descripción tendrá un punto de publicación, desde el cual un servidor donantonio lo incorpora a su biblioteca a través del proceso de registro.

Consulta y replicación de descripciones

Los servidores donantonio incluirán mecanismos o protocolos para admitir la replicación total o parcial de las descripciones solicitada por los clientes.

Para este fin, los servidores deberán poder atender consultas de descripciones, para lo que se definirán los correspondientes mecanismos o protocolos.

Los mecanismos y lenguajes de búsqueda implantados en los servidores deberán ser compatibles con los de los clientes.

Restricciones

El sistema será software libre (de acuerdo con la licencia GNU-GPL o similar) y deberán ser libres aquellos componentes que reutilice.

El sistema se diseñará según un modelo cliente/servidor.

El sistema basará sus comunicaciones en protocolos estándar de Internet.

Los distintos subsistemas deberán tener un diseño e implementación sencillos, independientes de la plataforma o el lenguaje de programación.

Los servidores deben ser capaces de atender consultas concurrentemente.

Suposiciones y dependencias

Suposiciones

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por el equipo de diseño atendiendo sugerencias de la comunidad. Cualquier petición de cambios en la especificación debe ser aprobada por el equipo de diseño y gestionada por el mismo.

Dependencias

Aunque la funcionalidad de servidor es autónoma, pues solo se comunica con otros clientes cuando éstos le hacen una petición, puede depender de otros servidores si, utilizando la funcionalidad de cliente, replica de otros servidores información para el mantenimiento de su propia biblioteca.

Nota: Repasar esto. En realidad sí hay comunicación, pero

Requisitos específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son **ESENCIALES**, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de testabilidad : dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

Nota: Es conveniente hacer la «matriz de requisitos», que confronta los requisitos con el método de verificación.

Requisitos funcionales

REQ01: Registro de descripciones

El servidor debe registrar descripciones de recursos (metadatos y el correspondiente localizador URL).

REQ02: Visibilidad de las descripciones

Todo servidor donantonio implementará mecanismos de consulta que permitan seleccionar desde un cliente un conjunto de las descripciones que éste dispone.

REQ03: Transportabilidad de las descripciones

Todo servidor donantonio permitirá la exportación de un conjunto seleccionado de descripciones.

REQ04: Selección de descripciones

Todo servidor implementará mecanismos o lenguajes que permitan a los clientes especificar una consulta de sus recursos almacenados.

REQ05: Independencia entre servidores

Los servidores serán independientes para poder dar servicio autónomamente a los clientes.

REQ06: Unicidad de las descripciones

Las descripciones deberán ser únicas en cada biblioteca mantenida en cada servidor.

Requisitos de interfaces externos

Interfaces de usuario

REQ07: Configurabilidad

El servidor donantonio contará con interfaces de usuario que permita administrar todas sus funciones.

Interfaces hardware

No se han definido.

Interfaces software

REQ08: Comunicación con otros módulos

La comunicación de los servidores donantonio con los clientes se realizará mediante protocolos basados en estándares que permitan cumplir los demás requisitos del sistema.

Interfaces de comunicación

REQ09: Protocolos estándares en Internet

Los servidores se comunicarán con otros módulos del sistema mediante protocolos estándares en Internet, siempre que sea posible.

Requisitos de rendimiento

REQ10: Tiempo de respuesta

Los tiempos de respuesta a las peticiones de los clientes deberán ser reducidos, considerando una red de comunicaciones eficiente.

REQ11: Concurrencia

El sistema deberá soportar atención concurrente a las consultas.

Requisitos de desarrollo

REQ12: Ciclo de vida.

El ciclo de vida elegido para desarrollar el sistema será el de prototipo evolutivo, de manera que se puedan incorporar fácilmente cambios y nuevas funciones.

Requisitos tecnológicos

REQ13: Normas bibliográficas

Los esquemas bibliográficos a usar en el servidor estarán expresados en RDF.

Atributos

REQ14: Software libre

El sistema será software libre y, por tanto, cualquier componente software que reutilice también deberá ser libre.

REQ15: Portabilidad

El sistema será diseñado portable. Deberá ser posible implantar aplicaciones donantonio en diversos lenguajes y plataformas.

REQ16: Mantenibilidad

El sistema es susceptible de ser ampliado. Por tanto deberá diseñarse fácilmente mantenible, aplicando para su desarrollo las metodologías que para ello sean precisas.

Otros requisitos

Apéndices

Licencia

Donantonio: Especificación de Requisitos Software

Copyright (C) 2000 Juan J. Amor, David Escorial e Ismael Olea.

Se otorga permiso para reproducir total o parcialmente este documento, siempre y cuando se cite la fuente. Cualquier modificación de este documento deberá ser aprobada por los autores.

Cientes Donantonio
Especificación de requisitos software

Juan José Amor

David Escorial

Ismael Olea

Tabla de contenidos

Introducción.....	121
Descripción general.....	123
Requisitos específicos	125
Apéndices	127

Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el subsistema cliente del sistema bibliográfico de publicación distribuida automática Donantonio. Todo su contenido ha sido elaborado teniendo en cuenta las necesidades observadas en la experiencia de los autores en publicación en Internet. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar IEEE *Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998*.

Propósito

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El documento va dirigido tanto al equipo de desarrollo, como a los integrantes de los proyectos de documentación libre y a la comunidad de posibles usuarios finales. Este documento será el canal de comunicación entre las partes implicadas, tomando parte en su confección miembros de cada parte. Esta especificación está sujeta a revisiones por las partes implicadas, especialmente por los potenciales usuarios, que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

Ámbito del sistema

Este sistema es requerido a raíz de la creciente complicación de las labores de publicación de documentación del software libre.

Se ha constatado la necesidad de un sistema informático que automatice las labores de publicación de documentos en Internet, de tal forma que se garantice la fácil localización de los mismos y una rápida puesta a disposición a los usuarios.

Usando normas bibliográficas puede generalizarse el uso y aplicación del sistema a cualquier tipo de información susceptible de ser clasificada.

El ámbito del sistema desarrollado llega hasta la realización de los clientes de Donantonio; prefiriéndose el diseño de un sistema que pueda ser empotrado en otras aplicaciones afines para que éstas puedan a su vez ser aplicaciones compatibles con otros módulos Donantonio.

Definiciones, acrónimos y abreviaturas

Definiciones

Recurso

Elemento accesible a través de una URL susceptible de ser clasificado bibliográficamente.

Esquema

Describe la sintaxis de una descripción.

Descripción

Instancia de un esquema que identifica mediante metadatos, de manera inequívoca a un recurso.

Biblioteca

Colección de descripciones.

Servidor

Nodo que atiende consultas donantonio.

Cliente

Nodo que genera consultas donantonio.

Consulta

Petición de un subconjunto de descripciones expresada en un lenguaje normalizado.

Acrónimos

ERS

Documento de Especificación de Requisitos Software.

GFDL

Licencia para Documentación Libre de GNU.

GPL

Licencia para Software Libre de GNU.

RDF

Resource Description Framework, estándar del consorcio de World Wide Web (W3C) para la codificación y manejo de metadatos usando el lenguaje de marcas XML.

URL

Uniform Resource Locator, esquema utilizado para la localización de un recurso en Internet.

XML

eXtended Markup Language. Lenguaje de marcas estructuradas, estándar del W3C.

Abreviaturas

No se han definido.

Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.
- Donantonio: sistema bibliográfico de publicación distribuida automática. Especificación de requisitos software. J.J. Amor, D. Escorial e I. Olea, 2000.
- Servidores Donantonio. Especificación de requisitos software. J.J. Amor, D. Escorial e I. Olea, 2000.

Visión general del documento

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema.

Descripción general

Perspectiva del producto

El cliente donantonio interactuará con servidores, mediante un protocolo que deberá definirse.

Funciones del sistema

Acceso a recursos

El cliente podrá obtener de los servidores donantonio la información que su administrador determine, en forma de conjuntos de descripciones de recursos (metadatos).

Los metadatos que obtienen los clientes describen el recurso, y éste no se encontrará necesariamente localizado en el servidor que proporcionó la descripción.

Biblioteca local

El cliente mantendrá, con las descripciones obtenidas de los servidores, una biblioteca local que se ajusta a los requisitos del administrador del cliente.

Consulta y replicación de descripciones

Los clientes donantonio incluirán mecanismos o protocolos para realizar una réplica total o parcial de las descripciones almacenadas en un servidor.

Los clientes, para este fin, elaborarán consultas en un determinado lenguaje que deberá definirse. El lenguaje de consulta permitirá obtener como respuesta una o más descripciones, en función del patrón de consulta utilizado.

Acceso del usuario a recursos

El cliente o la biblioteca del cliente deberá aceptar también consultas de información que interactivamente efectúe un usuario, a través de una aplicación tipo *interfaz de usuario donantonio*.

Acceso del usuario a configuración

El cliente podrá ser configurado desde una interfaz de usuario. Por configurar un cliente se entiende establecer de qué servidores va a obtener la información, y cómo (mediante qué consultas).

Características de los usuarios

Las interfaces de usuario que puedan requerirse para esta aplicación deberán ser de manejo intuitivo, fácil de aprender y sencillo de manejar. El sistema deberá presentar un alto grado de usabilidad.

Restricciones

El sistema será software libre (de acuerdo con la licencia GNU-GPL o similar) y deberán ser libres aquellos componentes que reutilice.

El sistema se diseñará según un modelo cliente/servidor.

El sistema basará sus comunicaciones en protocolos estándar de Internet.

Los distintos subsistemas deberán tener un diseño e implementación sencillos, independientes de la plataforma o el lenguaje de programación.

Nota: El requisito «Los clientes deben ser capaces de atender a varias peticiones de usuario simultáneamente» parece más que un requisito actual una interesante mejora...

Suposiciones y dependencias

Suposiciones

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por el equipo de diseño atendiendo sugerencias de la comunidad. Cualquier petición de cambios en la especificación debe ser aprobada por el equipo de diseño y gestionada por el mismo.

Dependencias

El cliente Donantonio requiere comunicarse con servidores para mantener su biblioteca. Por tanto hay una dependencia de la funcionalidad que ofrezcan éstos.

Requisitos específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son ESENCIALES, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de testabilidad : dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

Nota: Es conveniente hacer la «matriz de requisitos», que confronta los requisitos con el método de verificación.

Requisitos funcionales

REQ01: Selección de descripciones

Todo cliente donantonio implementará mecanismos de consulta que permitan acceder en un servidor a un conjunto de las descripciones que éste dispone.

REQ02: Replicación de descripciones

Todo cliente donantonio podrá replicar un conjunto de descripciones obtenidas desde un servidor donantonio mediante un criterio de selección de los definidos en REQ01.

REQ03: Almacenamiento de descripciones

Todo cliente tendrá una biblioteca local donde almacenará las descripciones que obtenga mediante consultas de los servidores.

REQ04: Unicidad de las descripciones

Las descripciones deberán ser únicas en la biblioteca de cada cliente.

REQ05: Programación de consultas

Todo cliente almacenará internamente las diferentes consultas que realizará periódicamente para actualizar el contenido de su biblioteca. La programación de consultas consistirá en ajustar toda la información necesaria para ejecutarlas (la propia consulta, el servidor contra el que se ejecuta y cuándo se ejecuta).

Requisitos de interfaces externos

Interfaces de usuario

REQ06: Interfaz de usuario.

El cliente donantonio podrá configurar todas sus funciones mediante una interfaz de usuario sencilla de utilizar.

Interfaces hardware

No se han definido.

Interfaces software

REQ07: Comunicación con otros módulos

La comunicación de los clientes donantonio con los servidores se realizará mediante protocolos basados en estándares que permitan cumplir los demás requisitos del sistema.

Interfaces de comunicación

REQ08: Protocolos estándares en Internet

Los servidores, clientes y aplicaciones donantonio se comunicarán entre sí mediante protocolos estándares en Internet, siempre que sea posible. Por ejemplo, para transferir ficheros (recursos o bloques de descripciones) deberán utilizarse protocolos existentes (FTP u otro conveniente).

Requisitos de rendimiento

REQ09: Tiempo de respuesta

Los tiempos de respuesta a las consultas de la biblioteca local procedentes de un interfaz de usuario deberán ser reducidos.

Requisitos de desarrollo

REQ10: Ciclo de vida.

El ciclo de vida elegido para desarrollar el sistema será el de prototipo evolutivo, de manera que se puedan incorporar fácilmente cambios y nuevas funciones.

Requisitos tecnológicos

REQ11: Normas bibliográficas

Los esquemas bibliográficos a usar en el sistema estarán expresados en RDF.

Atributos

REQ12: Software libre

El sistema será software libre y, por tanto, cualquier componente software que reutilice también deberá ser libre.

REQ13: Portabilidad

El sistema será diseñado portable. Deberá ser posible implantar aplicaciones donantonio en diversos lenguajes y plataformas.

REQ14: Mantenibilidad

El sistema es susceptible de ser ampliado. Por tanto deberá diseñarse fácilmente mantenible, aplicando para su desarrollo las metodologías que para ello sean precisas.

Otros requisitos

Apéndices

Licencia

Donantonio: Especificación de Requisitos Software

Copyright (C) 2000 Juan J. Amor, David Escorial e Ismael Olea.

Se otorga permiso para reproducir total o parcialmente este documento, siempre y cuando se cite la fuente. Cualquier modificación de este documento deberá ser aprobada por los autores.

Cientes Donantonio

Interfaz de usuario Donantonio
Especificación de requisitos software

Juan José Amor

David Escorial

Ismael Olea

Tabla de contenidos

Introducción.....	131
Descripción general.....	133
Requisitos específicos	134
Apéndices	137

Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el subsistema interfaz de usuario del sistema bibliográfico de publicación distribuida automática Donantonio. Todo su contenido ha sido elaborado teniendo en cuenta las necesidades observadas en la experiencia de los autores en publicación en Internet. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar IEEE *Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998*.

Propósito

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El documento va dirigido tanto al equipo de desarrollo, como a los integrantes de los proyectos de documentación libre y a la comunidad de posibles usuarios finales. Este documento será el canal de comunicación entre las partes implicadas, tomando parte en su confección miembros de cada parte. Esta especificación está sujeta a revisiones por las partes implicadas, especialmente por los potenciales usuarios, que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

Ámbito del sistema

Este sistema es requerido a raíz de la creciente complicación de las labores de publicación de documentación del software libre.

Se ha constatado la necesidad de un sistema informático que automatice las labores de publicación de documentos en Internet, de tal forma que se garantice la fácil localización de los mismos y una rápida puesta a disposición a los usuarios.

Usando normas bibliográficas puede generalizarse el uso y aplicación del sistema a cualquier tipo de información susceptible de ser clasificada.

El ámbito del sistema desarrollado llega hasta la realización de los interfaces de usuario de Donantonio; prefiriéndose el diseño de un sistema que pueda ser empotrado en otras aplicaciones afines para que éstas puedan a su vez ser aplicaciones compatibles con otros módulos Donantonio.

Definiciones, acrónimos y abreviaturas

Definiciones

Recurso

Elemento accesible a través de una URL susceptible de ser clasificado bibliográficamente.

Esquema

Describe la sintaxis de una descripción.

Descripción

Instancia de un esquema que identifica mediante metadatos, de manera inequívoca a un recurso.

Biblioteca

Colección de descripciones.

Servidor

Nodo que atiende consultas donantonio.

Cliente

Nodo que genera consultas donantonio.

Consulta

Petición de un subconjunto de descripciones expresada en un lenguaje normalizado.

Acrónimos

ERS

Documento de Especificación de Requisitos Software.

GFDL

Licencia para Documentación Libre de GNU.

GPL

Licencia para Software Libre de GNU.

RDF

Resource Description Framework, estándar del consorcio de World Wide Web (W3C) para la codificación y manejo de metadatos usando el lenguaje de marcas XML.

URL

Uniform Resource Locator, esquema utilizado para la localización de un recurso en Internet.

XML

eXtended Markup Language. Lenguaje de marcas estructuradas, estándar del W3C.

Abreviaturas

No se han definido.

Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.
- Donantonio: sistema bibliográfico de publicación distribuida automática. Especificación de requisitos software. J.J. Amor, D. Escorial e I. Olea, 2000.
- Clientes Donantonio. Especificación de requisitos software. J.J. Amor, D. Escorial e I. Olea, 2000.
- Servidores Donantonio. Especificación de requisitos software. J.J. Amor, D. Escorial e I. Olea, 2000.

Visión general del documento

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema.

Descripción general

Perspectiva del producto

La interfaz de usuario interactuará con clientes donantonio, mediante un mecanismo de comunicación que deberá definirse.

Funciones del sistema

Acceso a la información

La interfaz de usuario tendrá acceso a la biblioteca local de un cliente Donantonio, para consultar los metadatos allí contenidos.

Mecanismos de navegación

La interfaz de usuario incluirá mecanismos que faciliten la navegación por la información de un cliente Donantonio.

Nota: Ahora que lo pienso esta es la parte menos definida... probablemente haya que definir aquí las consultas interactivas y todo eso

Características de los usuarios

Las interfaces de usuario deberán ser de manejo intuitivo, fácil de aprender y sencillo de manejar. El sistema deberá presentar un alto grado de usabilidad. Lo deseable sería que un usuario nuevo se familiarizase con el sistema en muy poco tiempo, siendo de uso inmediato para las funciones de búsqueda de recursos.

Restricciones

El sistema será software libre (de acuerdo con la licencia GNU-GPL o similar) y deberán ser libres aquellos componentes que reutilice.

El sistema se diseñará según un modelo cliente/servidor.

El sistema basará sus comunicaciones en protocolos estándar de Internet.

Los distintos subsistemas deberán tener un diseño e implementación sencillos, independientes de la plataforma o el lenguaje de programación.

Suposiciones y dependencias

Suposiciones

Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por el equipo de diseño atendiendo sugerencias de la comunidad. Cualquier petición de cambios en la especificación debe ser aprobada por el equipo de diseño y gestionada por el mismo.

Dependencias

La interfaz de usuario Donantonio realiza sus consultas sobre la biblioteca perteneciente a un cliente Donantonio. Por tanto, hay una dependencia de comunicación entre ambos.

Requisitos específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son ESENCIALES, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de testabilidad : dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

Nota: Es conveniente hacer la «matriz de requisitos», que confronta los requisitos con el método de verificación.

Requisitos funcionales

REQ01: Acceso a biblioteca

La interfaz de usuario podrá acceder a toda la biblioteca de un cliente donantonio.

REQ02: Búsquedas sobre la biblioteca

La interfaz de usuario permitirá realizar búsquedas de descripciones, para lo cual se pueden especificar total o parcialmente las descripciones a buscar.

REQ03: Refinamiento

La interfaz de usuario, tras efectuar una búsqueda y mostrar los resultados al usuario, permitirá refinar la búsqueda, es decir, reescribirla a partir de la actual haciéndola más restrictiva.

REQ04: Acceso a recursos

Los resultados de la búsqueda incluirán mecanismos que permitirán al usuario solicitar la realización de una copia local del recurso referenciado por la descripción.

REQ05: Programación del cliente

El usuario contará con mecanismos para programar su cliente, es decir, introducir en éste las consultas donantonio que efectuará contra los servidores elegidos, y cuándo se efectúan.

Requisitos de interfaces externos

Interfaces de usuario

REQ06: Requisito general de la interfaz de usuario

La interfaz de usuario diseñada deberá tener alto grado de usabilidad y ser sencilla, cumpliendo todos los requisitos especificados.

Interfaces hardware

No se han definido.

Interfaces software

REQ07: Comunicación con otros módulos

La comunicación con los clientes donantonio se realizará mediante protocolos basados en estándares que permitan cumplir los demás requisitos del sistema.

Interfaces de comunicación

No se han definido

Requisitos de rendimiento

REQ09: Consumo de recursos

La complejidad de las aplicaciones de usuario debe ser baja, y consumir pocos recursos.

Requisitos de desarrollo

REQ10: Ciclo de vida.

El ciclo de vida elegido para desarrollar el sistema será el de prototipo evolutivo, de manera que se puedan incorporar fácilmente cambios y nuevas funciones.

Requisitos tecnológicos

REQ11: Normas bibliográficas

Los esquemas bibliográficos a usar en el sistema estarán expresados en RDF.

Atributos

REQ12: Software libre

El sistema será software libre y, por tanto, cualquier componente software que reutilice también deberá ser libre.

REQ13: Portabilidad

El sistema será diseñado portable. Deberá ser posible implantar aplicaciones donantonio en diversos lenguajes y plataformas.

REQ14: Mantenibilidad

El sistema es susceptible de ser ampliado. Por tanto deberá diseñarse fácilmente mantenible, aplicando para su desarrollo las metodologías que para ello sean precisas.

Otros requisitos

taca taca

Apéndices

Licencia

Donantonio: Especificación de Requisitos Software

Copyright (C) 2000 Juan J. Amor, David Escorial e Ismael Olea.

Se otorga permiso para reproducir total o parcialmente este documento, siempre y cuando se cite la fuente. Cualquier modificación de este documento deberá ser aprobada por los autores.

Un día en la vida de un recurso en la red Donantonio

Ismael Olea

Nota: esto es sólo una visión parcial que pretende ser ilustrativa. Casi nada para estos tiempos.

1. Un recurso es publicado en un servidor inet (con una URL)
2. Creación de la descripción bibliográfica del recurso
 - creación manual
 - asistida por un editor de metadatos
 - semiautomática, mediante un programa de extracción de palabras clave.
3. Publicación de la descripción en inet (con una URL)
4. Registro de la descripción en un servidor Donantonio (Sugerencia: probablemente el servidor debe distinguir entre:
 - descripciones revisadas
 - descripciones no revisadas
 - descripciones copiadas de otros servidores Donantonio)
5. El servidor Donantonio:
 - archiva una copia de la descripción en su almacén público
 - alimenta con la descripción su sistema de gestión de datos.
6. El servidor Donantonio atiende consultas Donantonio
7. El cliente Donantonio que lanzó la consulta recoge del servidor consultado como resultado una serie de descripciones que la cumplen.
8. El cliente Donantonio puede archivar el resultado de la consulta en su propio almacén o bien puede acceder al recurso/s vía URL.

Nota: ¿cómo lo véis? ¿os aclara algo? :-?

Aplicaciones de donantonio

Ismael Olea

Ejemplos de aplicaciones donantonio

Tabla de contenidos

Dorsales	145
Replicación de recursos	145
Tolerancia a caídas de la red	145

Dorsales

Pueden construirse redes dorsales que garanticen la disponibilidad de todas las descripciones registradas en una red donantonio:

Dorsal de Donantonio

red de servidores en la que siempre es posible localizar cualquier recurso dado de alta en Donantonio.

Servidor genérico

servidor donantonio que forma parte de la dorsal de la red.

Servidor especializado

servidor donantonio que replica solo una parte de las descripciones que guarda la dorsal de la red donantonio, y por tanto no puede formar parte de dicha dorsal.

Replicación de recursos

El sistema permitirá, mediante una selección previa (vía consulta), realizar una replicación local de los recursos seleccionados.

De esta forma, mediante protocolos estándar Internet y otros específicamente diseñados para este sistema, se podrá realizar una copia local de los recursos que resulten de nuestro interés (por ejemplo, cumplan un criterio de búsqueda taxonómico determinado) con el fin de crear una biblioteca-e local.

Tolerancia a caídas de la red

El diseño de la red de servidores dará un matiz de fiabilidad al sistema. Es decir, una caída de un servidor no impedirá necesariamente la localización de un recurso, cuando su descripción se encuentre replicada en otros servidores.

