

Biblioteca Digital
Universidad Nacional de Cuyo

Necesidades de la Universidad

- Contar con un sistema de administración para la biblioteca digital que permita:
 - Manejo jerárquico de distintos tipos de objetos.
 - Administración distribuida y concurrente entre distintas áreas de la universidad.
 - Portal único para todos los objetos digitales.
 - Adaptado a las capacidades tecnológicas de la universidad

Respuestas buscadas en otros sistemas

- En la administración:
 - Flexibilidad y facilidad de creación de plantillas de metadatos.
 - Facilidad de carga de metadatos asistidos por listas de palabras definibles.
 - Definición jerárquica de los objetos
 - Distintos niveles de usuarios y concurrencia durante la carga.

Respuestas buscadas en otros sistemas

- En la visualización de los objetos
 - Flexibilidad a la hora de desarrollar las plantillas públicas.
 - Cada tipo de archivo debe tener su propia plantilla de visualización con detalle de licencias de uso y relaciones con otros objetos.
 - Generación de estadísticas de utilización.
 - Libertad de incorporar nuevos elementos como formularios de consulta o integración con otros buscadores.

Luego de la búsqueda la **decisión**

- **Desarrollar un sistema propio** sin cerrar la puerta a futuras migraciones a otros sistemas.
- Comenzar con las pruebas de carga y administración de la biblioteca digital.
- Entrenar a los administradores en el uso de herramientas de conversión de objetos y publicación en la Web.

Diseño de la aplicación

- Se decidió continuar con la política de la universidad de utilizar software libre para el desarrollo.
- Se eligió PHP como lenguaje de programación y Postgresql como motor de bases de datos, el sistema operativo de base es Linux.

Primeras decisiones importantes

- Qué formatos utilizar para la publicación ?
 - Audio: mp3, utilización de reproductores basados en flash.
 - Video: flv, originales en mpg (alta calidad), difícil compresión.
 - Textos: pdf, sin mayores complicaciones.
 - Imágenes: jpg, sin mayores complicaciones

En general se ha utilizado Flash como reproductor por la alta tasa de instalación de este plugin en los navegantes (ca. 95% de las pcs)

http://www.alzado.org/articulo.php?id_art=438

Metadatos

- Se utiliza Dublin Core, más campos locales para la definición de cada objeto.
- Se discute cada campo con los responsables de contenido y se adaptan las plantillas públicas para mostrar los nuevos cambios.

ESTAMOS APRENDIENDO!!!

Herramientas utilizadas

PHP5

Lenguaje del servidor

POSTGRESQL

Base de datos

Javascript / ajax

En el cliente

Estructuras de datos



Estructura básica del sistema

Tipo de objeto

Plantilla

Metadatos

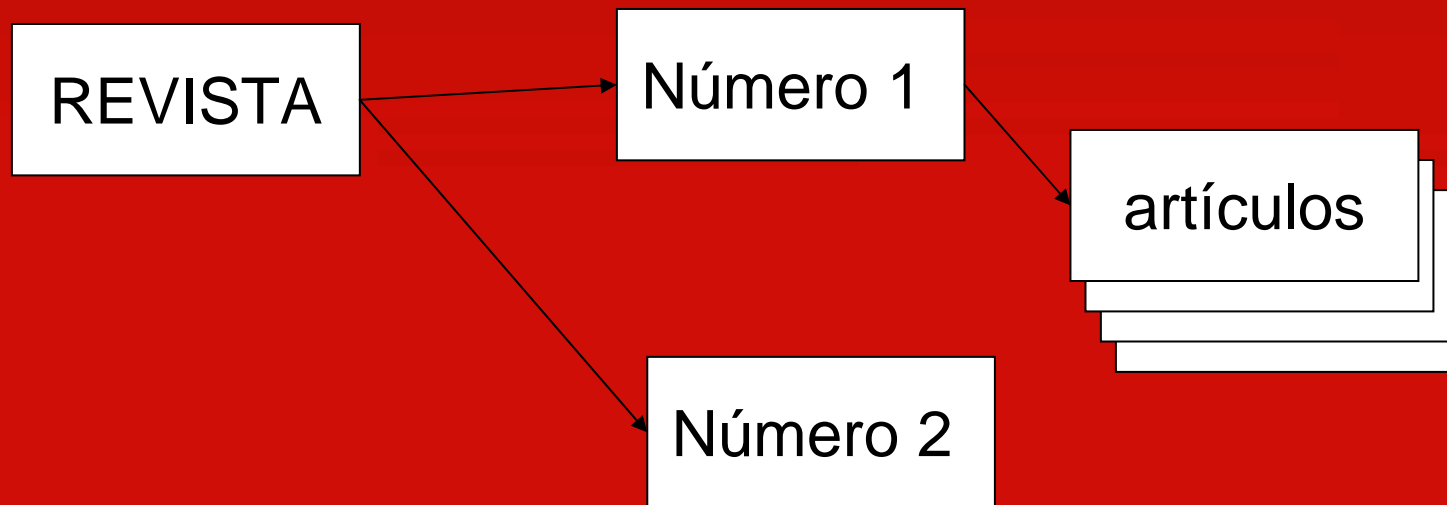
Video / flv

Bloque de Video

dc.subject

Jerarquías de objetos

- Los objetos pueden tener objetos dependientes o depender de otros objetos. Las asociaciones se definen mediante metadatos locales (por ej: asoc_video) .



El Objetivo principal de utilizar este concepto es poder **navegar** la colección

Dependencia entre metadatos

También se puede definir relaciones entre metadatos

Metadato principal

dc.creator

Submetadatos

e-mail

filiación

Campo interno
que determina la
Unidad
académica a la
que pertenece

Especialmente utilizados en la visualización de las fichas de los objetos.

Secciones del sistema desarrolladas

- Definición de formatos de objetos
 - pdf, jpg, mp3, flv, etc. Definen el tipo de archivo y como tratarlo.
- Definición de tipos de objetos:
 - revistas, libros, videos, etc.. Definen el tipo de objeto, permite luego buscar asociaciones.
- Definición de la plantilla de carga.
 - Se definen las relaciones con otros objetos y los metadatos necesarios.
- Definición de plantillas de publicación para cada tipo de objeto (aislar el trabajo del diseñador del programador)

Opciones de administrador

- Definición de tipos de objetos:
 - Tipo de objeto, plantilla pública, relaciones con otros objetos.
- Definición de plantillas de carga:
 - Definición de los metadatos a utilizar y su forma de presentación (texto, textarea, listas seleccionables etc.), si está asociado a una lista, si es dependiente de otro metadato.

Publicación de objetos

1. Preparar el objeto digital, compresión, edición, etc.
2. Subida del objeto digital ya procesado.
3. Carga de los metadatos utilizando las listas desplegadas definidas, listas automáticas.
4. Definir la relación con otros objetos.
5. Visualizar la ficha pública en modo administrador.
6. Publicación definitiva.

Diseño de las plantillas públicas

- El objetivo es permitir al diseñador independencia a la hora de crear las plantillas públicas.
- Se le proveen “marcas html” para colocar los metadatos y funciones de conversión para tratar los metadatos.

Ejemplos de marcas

- `{$dc.date|formatofecha::Y}`
metadato original : 2006-10-19
se presenta en la plantilla : 2006
recupera solo el año
- `{$dc.title|occ::1}`
recupera solo la primera ocurrencia del título
- `{$idobjeto|hijos::BLOQUE}`
Buscará todos los hijos que dependen del objeto “idobjeto” y los mostrará utilizando el formato especificado en BLOQUE.

Pendientes

- Mejorar las opciones de búsqueda en la sección pública. (tsearch2, lucene)
- Normalizar los nombres de objetos y funciones de plantilla
- Generar nuevos tipos de campos para la carga.

Y muchas otras cosas...