

# REPASA, un Repositorio Web de objetos de aprendizaje estandarizados por taxonomías



Mónica Cristina Bazán Milla  
Carlos Sánchez Juanes  
Iván del Valle Gómez

**Director: Antonio Sarasa Cabezuelo**

Sistemas Informáticos  
Curso 2010-2011  
Facultad de Informática  
Universidad Complutense de Madrid



# Índice

Autorización.....	5
Capítulo 1: Resumen y palabras clave .....	7
1.1 Versión en castellano .....	7
1.2 English version .....	7
Capítulo 2: Introducción.....	9
Capítulo 3: Estado del arte .....	13
Capítulo 4: Repositorio Web RepASA .....	19
Capítulo 5: Arquitectura de la aplicación .....	21
5.1 Casos de uso.....	21
5.2 Navegación por las páginas PHP .....	23
5.2.1 Usuario no registrado (página Index).....	23
5.2.2 Administrador.....	23
5.2.3 Usuario registrado .....	25
5.3 Inclusiones entre archivos PHP .....	26
5.4 Estructura de la base de datos.....	26
5.5 Ciclo de vida de un archivo.....	32
Capítulo 6: Funcionalidad de la aplicación.....	33
6.1 Generalidad .....	35
6.2 Accesibilidad.....	36
6.3 Manual de explotación del administrador .....	39
6.4 Manual de explotación del usuario registrado.....	52
6.5 Manual de explotación del usuario no registrado .....	68
Capítulo 7: Dificultades, conclusiones y trabajo futuro .....	75
Bibliografía.....	77
Anexo 1: Guía de instalación de XAMPP y Repositorio.....	79
Anexo 2: Ejemplo de taxonomía .....	87
Anexo 3: Normas de Accesibilidad Web.....	95



## **Autorización**

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Fdo.:

Mónica Cristina Bazán Milla:

Carlos Sánchez Juanes:

Iván del Valle Gómez:



# Capítulo 1: Resumen y palabras clave

## *1.1 Versión en castellano*

El repositorio RepASA de material didáctico es una aplicación Web cuyo principal objetivo es el de facilitar a los alumnos y profesores de diferentes materias, compartir y acceder a material educativo que les ayude en sus estudios.

A diferencia de otros recursos similares, este repositorio tiene la ventaja de que son sus propios usuarios registrados quienes, mediante sus propias subidas de material, se encargan de actualizar y de controlar sus contenidos, incluso los de otros usuarios.

Nuestro objetivo principal ha sido crear una aplicación útil para todo tipo de estudiantes y profesores a la hora de compartir material didáctico de diferentes campos, además de que sea fácil de buscar y encontrar lo que se necesite, incluso donde gente no registrada también puede aprovechar lo que se publica en el repositorio.

**Palabras clave:** Repositorio, Web, aprendizaje, taxonomía, LOM, etiquetas, archivos, profesores, alumnos, almacenamiento.

## *1.2 English version*

The RepASA repository of teaching materials is a web application whose main purpose is to provide students and teachers of different subjects, share and access to educational materials to help them in their studies.

Unlike other similar resources, this repository has the advantage that they are their own registered users who, through their own material upload, are responsible for updating and controlling its content, including by other users.

Our main goal has been to create a useful application for all students and teachers in sharing teaching materials in different fields, as well as make it easy to search and find what you need, even no registered people can also benefit of what published in the repository.

**Keywords:** Repository, Web, learning, taxonomy, LOM, tags, files, teachers, students, storage.



## Capítulo 2: Introducción

El término repositorio proviene del latín *repositorium* [3] y significa armario, alacena, es decir, lugar donde se guarda algo. Por tanto, podemos definir un repositorio como un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos, bajo la misma ubicación. Es decir, son lugares donde se almacenan colecciones de recursos digitales de forma organizada con un sistema descriptivo a través de metadatos. Los metadatos, son un conjunto de atributos que sirven para definir la información que se almacena dentro de un recurso de manera que se pueda ser catalogada y categorizada.

La existencia de los repositorios surge principalmente por la necesidad de almacenar y encontrar de forma fácil y segura la inmensa cantidad de información que actualmente existe en Internet. Además, las instituciones están generando un número elevado, complejo y heterogéneo de contenidos digitales. También aparece como respuesta a la falta de normalización de los contenidos.

El objetivo principal de un repositorio es reunir, preservar y dar acceso a la producción de los investigadores y docentes en múltiple formatos. Es un sistema en línea de acceso abierto, es decir, se apoyan en Internet para facilitar el acceso y la difusión de sus contenidos a sus usuarios.

La importancia de los repositorios radica en la fiabilidad de los contenidos, pues los repositorios disponen de sistemas de evaluación y filtrado, bien a manos de los administradores o bien, mediante los propios usuarios del repositorio.

En los repositorios es necesario pues definir un tipo de clasificación [5] para los archivos. Existen en la actualidad cuatro tipos de clasificación posibles:

### ❖ Taxonomías

Se trata de un conjunto de términos entre los que se establecen relaciones jerárquicas. Las jerarquías definen implícitamente un árbol conceptual, de manera que la clasificación consiste en encontrar el camino taxonómico dentro del árbol, que va desde la raíz hasta la hoja que representa al término que mejor describe al recurso.

Es decir, consisten en un sistema de clasificación jerárquico en base a *categorías* semánticas de un campo determinado del conocimiento que se ramifican en árboles de vocablos denominados *taxones*.

Cada subárbol asociado a una categoría trata de definir de la forma más precisa posible el concepto representado en la categoría padre mediante otros vocablos relacionados con él por una relación padre-hijo o de composición.

Así para clasificar un recurso se recorre el árbol taxonómico, y se le asocian todos aquellos caminos de taxones que de acuerdo a la naturaleza del recurso, mejor lo definen.

Nuestro proyecto está centrado en el desarrollo de un repositorio cuya clasificación es por taxonomías.

#### ❖ Tesauros

Se trata de un conjunto de términos entre los que se establecen relaciones semánticas y genéricas: de equivalencia, jerárquicas y asociativas. Estas relaciones definen implícitamente un grafo conceptual, en el que se pueden definir a su vez subgrafos de conceptos que se denominan *microtesauros*. La clasificación normalmente consiste en encontrar uno o más caminos no cíclicos dentro de los subgrafos que representan a los microtesauros.

#### ❖ Vocabularios libres

Son un conjunto de palabras definidas (también llamadas *tags* o etiquetas) por los usuarios del repositorio las cuales no se rigen por ningún tipo de estructura jerarquizada. Son creados por los usuarios como un trabajo colaborativo, pues son los propios usuarios los encargados de etiquetar los recursos. También son conocidas como *folksonomías*.

#### ❖ Vocabularios controlados

Se trata de una lista de términos relacionados, pero sin relaciones semánticas entre los mismos. La forma de clasificar consiste en asociar a los recursos uno o más términos de la lista, aquellos que mejor describen el recurso.

Consiste en un conjunto de vocablos que son usados para caracterizar un recurso. Así para llevar a cabo su clasificación, se toma el recurso, y se eligen cuáles de esos vocablos caracterizan mejor al recurso, quedando asociados al mismo. De esta forma cuando se realiza una búsqueda, el recurso aparecerá siempre que entre las condiciones de búsqueda se utilice alguna de los vocablos que se le han asociado.

A pesar de la gran diversidad de repositorios digitales que hoy en día podemos encontrar, hay un conjunto de características comunes en la arquitectura de cualquier sistema de información, como son en este caso los repositorios:

- **Colección:** desarrollo y gestión de colecciones de recursos digitales, locales o distribuidos, sin restricción de formato.

- **Servicios de valor añadido:** productos y facilidades creados para dar valor al contenido de la colección, adecuados a las necesidades y a los requisitos de sus usuarios.
- **Personalización:** funcionalidad para que el usuario (o institución) pueda definir su espacio de interacción con el repositorio digital y seleccionar en listas propias los elementos de la colección.
- **Ciclo de vida de la información:** los contenidos digitales pueden tener fases diferentes en sus diversas etapas y debe llevarse un seguimiento del ciclo de cada recurso.



## Capítulo 3: Estado del arte

### REPOSITORIOS DIGITALES (ACTUALES)

**Repositorio digital [9]**, es un sistema para almacenar, conservar y dar acceso a documentos digitales. Más concretamente, es un conjunto de servicios que una o varias instituciones ofrecen a los miembros de su comunidad para la gestión y difusión de los materiales digitales creados por la Institución y su comunidad de miembros. Esencialmente supone un compromiso de organización para el control de esos contenidos (políticas) incluyendo su preservación en el tiempo así como su organización y acceso o distribución con el fin de maximizar la visibilidad e impacto en línea.

Podemos clasificar los repositorios atendiendo al tipo de acceso: públicos o protegidos.

- Públicos, puede acceder cualquier persona.
- Protegidos, es necesario autenticación previa.

Los depósitos más conocidos son los de carácter académico e institucional y tienen por objetivo organizar, archivar, preservar y difundir la producción intelectual resultante de la actividad investigadora de la entidad.

### REPOSITORIOS EDUCATIVOS

En el ámbito de los entornos virtuales, podríamos considerar como objetos de aprendizaje a aquellos archivos o unidades digitales de información dispuestos con la intención de ser utilizados en diferentes propuestas y contextos pedagógicos. Más concretamente, nos referimos a archivos digitales o elementos con un cierto nivel de interactividad e independencia, que podrían utilizarse en diferentes situaciones de enseñanza-aprendizaje y que deberían disponer de las indicaciones suficientes para su referencia e identificación. Los objetos de aprendizaje pueden adquirir formas muy diversas y presentarse en diferentes formatos y soportes.

Dado que son muchos los estudios realizados al respecto en los últimos años, algunos autores organizan estos objetos en diferentes niveles, al margen de las múltiples agrupaciones que, de acuerdo con los campos de los descriptores, podríamos configurar. En el caso de la estructuración por niveles, el primero de ellos haría referencia a las unidades más pequeñas a las que podríamos denominar objetos de aprendizaje: una imagen digital, una tabla, frase, fórmula o sonido. Los siguientes niveles van suponiendo unidades cada vez más complejas y, lógicamente, menos adaptables a otros contextos o contenidos de aprendizaje. Como ya hemos dicho, de acuerdo con los campos, áreas del saber, y otras formas taxonómicas, la clasificación de los objetos puede adquirir diferentes formas.

Para esa organización, estos objetos, además de su característica de reutilización, han de contar con la posibilidad de ser actualizados, combinados, referenciados y sistematizados. Así podremos catalogarlos y etiquetarlos para ser ubicados en los diferentes repositorios, con el fin de que posteriormente puedan ser localizados para su reutilización o, si procede, modificación o reelaboración, mediante las correspondientes estrategias de contraste, comparación y crítica de la información obtenida. Por eso resulta obvia la necesidad de potentes repositorios de objetos de aprendizaje. De ahí que el objeto y el repositorio sean dos entes complementarios. Un objeto que no posea las características necesarias para poder integrarse en un repositorio pierde todas sus virtudes y a la vez, un repositorio que no cuente con una buena base de objetos, deja de ser interesante y operativo.

Alguna de las características sistematizadas de los objetos de aprendizaje son las siguientes:

- **Reutilización:** objeto con capacidad para ser usado en contextos y propósitos educativos diferentes.
- **Educatividad:** capacidad para generar aprendizaje.
- **Interoperabilidad:** capacidad para poder integrarse en estructuras (plataformas) diferentes.
- **Accesibilidad:** facilidad para ser identificados, buscados y encontrados gracias al correspondiente etiquetado a través de diversos descriptores (metadatos) que permitirían la catalogación y almacenamiento en el correspondiente repositorio.
- **Durabilidad:** vigencia de la información de objetos, sin necesidad de nuevos diseños.
- **Independencia y autonomía:** de los objetos respecto a los sistemas donde fueron creados.
- **Generatividad:** capacidad para construir contenidos y objetos nuevos derivados de él. Capacidad para ser modificados o actualizados, aumentando su potencia a través de la colaboración.
- **Flexibilidad, versatilidad y funcionalidad:** para combinarse en diferentes áreas del conocimiento.

Existen repositorios institucionales, de empresas, de organizaciones, etc. La propia Web podría ser considerada como un gran repositorio, siempre que les aplicásemos las estrategias de búsqueda, procesamiento, selección y catalogación a través de esquemas de metadatos. La estructura de metadatos supone contar con una detallada estructura textual, que describe atributos, propiedades y características distribuidos en diferentes casos que identifican claramente al objeto, con el fin de que pueda encontrarse y utilizarse.

Por tanto, entendemos como repositorios de objetos de aprendizaje, una gran colección de los mismos, estructurada como un banco o base de datos con metadatos asociados y que generalmente podemos buscar en los entornos Web. La utilización de metadatos facilita la indexación de objetos que así podrían ser buscados sin problemas en internet.

Los elementos que están guardados dentro de los repositorios de los contenidos digitales, suelen estar catalogados mediante un conjunto de atributos que vienen definidos por los metadatos. Estos metadatos (palabra compuesta por el lexema meta (a cerca de) y el lexema datos) son un conjunto de atributos o elementos necesarios para describir un recurso. Así mismo sirven para definir los datos que forman parte de un objeto.

Actualmente existen diversos tipos de metadatos:

#### ❖ Dublin Core [8]

Estos metadatos son de tipo descriptivo. Fueron creados inicialmente para catalogar y compartir información sobre libros entre bibliotecas. Este modelo de metadatos está auspiciado por la DCMI (Dublin Core Metadata Initiative), que se dedica a fomentar la adopción extensa de los estándares interoperables de los metadatos. Dublin Core está definido por la norma ISO 15836 del año 2003, y la norma NISO Z39.85-2007.

La especificación de los elementos que hace Dublin Core está formada por las siguientes etiquetas:

- *DC.Title* (Título): El nombre dado al recurso.
- *DC.Subject* (Materias y palabras clave): El tema del contenido del recurso.
- *DC.Description* (Descripción del contenido del recurso): Puede incluir un resumen, una tabla de contenidos, etc.
- *DC.Source* (Fuente): Referencia al recurso del que deriva el documento actual.
- *DC.Lenguaje* (Lengua): El idioma del contenido del recurso.
- *DC.Relation* (Relación): Una referencia a un recurso relacionado con el contenido.
- *DC.Coverage* (Cobertura): Ámbito del contenido del recurso. Puede tratarse de una especificación geográfica, temporal o legal.
- *DC.Creator* (Autor): Responsable de la creación del contenido. Puede ser una entidad, una persona o un servicio.
- *DC.Contributor* (Colaborador): Responsable de hacer colaboraciones al contenido del recurso.
- *DC.Rights* (Derechos): Información sobre los derechos de la propiedad intelectual del recurso.
- *DC.Date* (Fecha): Fecha asociada a la creación o modificación del recurso. Se suele seguir la notación AAAA-MM-DD.
- *DC.Type* (El tipo o categoría del contenido): Palabras clave de un vocabulario que describen la naturaleza del recurso.

- DC.Format (Formato): Descripción física del recurso, como su tamaño, duración, dimensiones, etc. si son aplicables. Se suelen usar tipos MIME.
- DC.Identifier (Identificación): Referencia unívoca para el contenido del recurso.

#### ❖ LOM [2]

Es un modelo de datos, codificado en XML [7], se utiliza para describir objetos de aprendizaje y otros recursos digitales usados para el apoyo al aprendizaje. Su propósito es facilitar la interacción con sistemas de aprendizaje en línea. En España su variante es LOM-ES.

LOM está estructurado en las siguientes categorías y elementos:

- General: agrupa la información general
  - Identificador, Título, Entrada de catálogo, Lengua, Descripción, Descriptor, Cobertura, Estructura, Nivel de agregación.
- Ciclo de vida: describe la historia y el estado actual de un objeto
  - Versión, Estatus, Otros colaboradores.
- Meta-metadatos: describe el propio registro de los metadatos
  - Identificador, Entrada de catálogo, Otros colaboradores, Esquema de metadatos, Lengua.
- Técnica: describe los requisitos técnicos de los objetos de aprendizaje.
  - Formato, Tamaño, Ubicación, Requisitos, Comentarios sobre la instalación, Otros requisitos para plataformas, Duración
- Uso educativo : describe las características pedagógicas del objeto de aprendizaje
  - Tipo de interactividad, Tipo de recurso de aprendizaje, Nivel de interactividad, Densidad semántica, Usuario principal, Contexto [Nivel educativo], Edad, Dificultad, Tiempo previsto de aprendizaje, Descripción, Lengua
- Derechos: describe los derechos de propiedad intelectual y las condiciones de uso.
  - Coste, Copyright y otras restricciones, Descripción
- Relación: describe las relaciones existentes entre un objeto de aprendizaje y otros
  - Tipo [naturaleza de la relación con el recurso principal], Recurso [recurso principal al que se refiere esta relación].
- Anotación: proporciona comentario sobre el uso pedagógico del objeto
  - Persona, Fecha, Descripción.
- Clasificación: describe donde se sitúa el objeto de aprendizaje dentro de un sistema de clasificación concreto.
  - Finalidad, Nivel taxón (taxonómico), Descripción, Descriptor.

Lo deseable es, que tanto los objetos como los repositorios se atengan a determinados procesos de estandarización con el fin de hacer posibles los intercambios, migración y encaje entre objetos entre repositorios y plataformas distintos. Existen varios estándares para objetos de aprendizaje. Si se llegase al consenso suficiente para que los estándares de objetos, repositorios y metadatos fuesen aceptados por los grandes grupos que trabajan en estos temas, se facilitaría en gran medida el intercambio de objetos entre repositorios diferentes y la interoperabilidad de los sistemas y plataformas, con lo que se incrementaría, sin duda, la expansión y calidad de los sistemas digitales de enseñanza y aprendizaje (*e-learning*).

## EJEMPLOS DE REPOSITARIOS

### ❖ **AGREGA [1]**

El repositorio Agrega ha sido desarrollado en España, en concreto este programa ha sido desarrollado y ejecutado por el Ministerio de Industria, Turismo y Comercio, la entidad pública empresarial Red.es, el Ministerio de Educación, Política Social y las Consejerías de Educación de las distintas Comunidades Autónomas de España. El objetivo de Agrega es facilitar a la comunidad educativa una herramienta útil para integrar las Tecnologías Informáticas en el aula. Se podrá acceder a cualquier contenido, desde cualquier lugar, utilizando distintos criterios de búsqueda.

Los elementos que almacena Agrega están guardados en nodos autonómicos. Agrega cataloga los elementos del repositorio según el estándar LOM-ES, el cuál añade tres nuevas etiquetas a LOM que son “proceso cognitivo” en la categoría 5, “tipo de acceso” y “descripción” que forman parte de la categoría número 6.

### ❖ **MERLOT [10]** (Multimedia Educational Resources for Learning an Online Teaching)

Sin duda es el repositorio más reconocido, es quizá el que está marcando la pauta para el desarrollo y tendencia de los ROA. Es un repositorio centralizado que contiene sólo los metadatos y apunta a los objetivos ubicados en sitios remotos. Es independiente y funciona como un portal de OA. Provee búsquedas y otros servicios como personalización, importación y exportación y de objetos. Cualquier usuario puede tener acceso a todos los objetos contenidos en MERLOT y sólo los miembros contribuyen agregando objetos, pero para ser miembros no se requiere más que inscribirse y no se adquiere ninguna responsabilidad. La revisión por pares es una actividad que MERLOT utiliza para evaluar la calidad de los objetos agregados.

#### ❖ FEDORA [11]

El repositorio de contenidos digitales Fedora es un proyecto originalmente desarrollado por la Universidad Cornell y la biblioteca de la Universidad de Virginia.

Los objetos digitales que almacena Fedora pueden estar guardados dentro del repositorio o estar guardado externamente y simplemente estar referenciados por el objeto digital. Estos objetos digitales están formados por un identificador único, unas propiedades del objeto que sirven para gestión y seguimiento del objeto, y un conjunto de datastreams que tienen un identificador único dentro del alcance del objeto digital. Además Fedora reserva cuatro de estos identificadores para datastreams que van a servir para definir la información que guarda el objeto.

#### ❖ CAREO [12] (Campus Alberta Repository of Educational Objects)

Es un repositorio centralizado de objetos de aprendizaje multidisciplinares de profesores de Alberta (Canadá). Es un repositorio independiente que da acceso a objetos remotos y locales a través de los metadatos contenidos en su colección. Cualquier usuario puede tener acceso a los objetos, pero los miembros tiene servicios adicionales, al igual que MERLOT ser miembro es gratis y abierto a cualquier persona.

## Capítulo 4: Repositorio Web RepASA

Como ya hemos mencionado anteriormente el repositorio RepASA de material didáctico es una aplicación Web cuyo principal objetivo es el de facilitar tanto a los alumnos y como profesores de diferentes materias, compartir y acceder a material educativo que les ayude en sus estudios.

RepASA, el repositorio y Almacén de Sabiduría ASA (es un acrónimo recursivo), debe su nombre a que se trata de un repositorio educativo donde poder encontrar archivos interesantes que estudiar y repasar, y, a la vez, un lugar donde se almacenan conocimientos y sabiduría en formato digital.

Este repositorio está a disposición de todo cualquier persona pues y resulta de especial interés tanto para estudiantes como para profesores. Además para acceder al mismo y realizar búsquedas no es necesario estar registrado.

Destacamos también la facilidad que tiene los usuarios de RepASA tanto para la subida de material didáctico como para la búsqueda del mismo dentro del repositorio.

Otro punto a destacar es que a diferencia de otros recursos similares, los encargados de actualizar y controlar todos los contenidos pertenecientes al repositorio son los propios usuarios registrados mediante la subida de material.

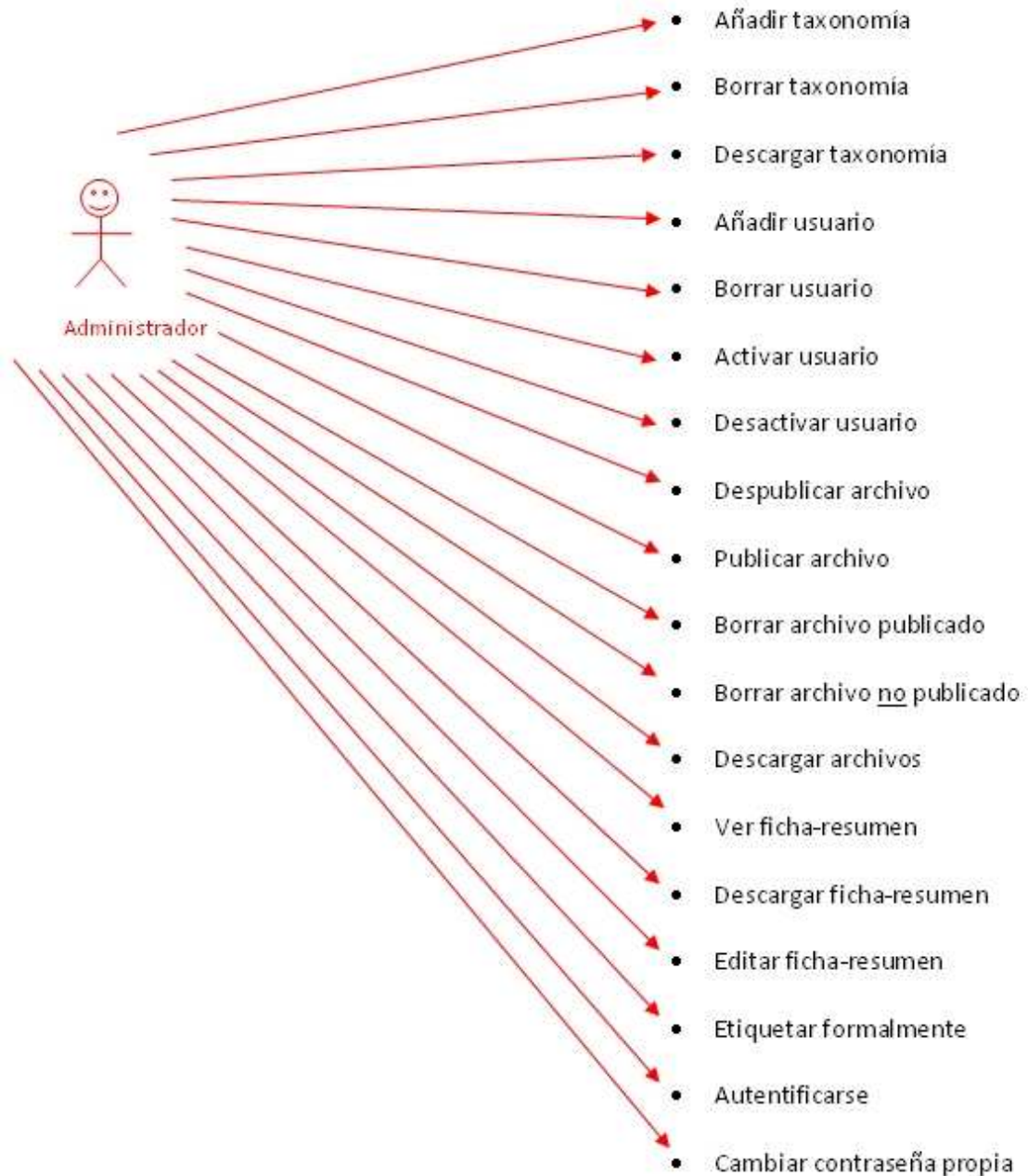
A continuación, pasaremos a detallar en más profundidad la arquitectura de la aplicación haciendo foco sobre todo en los todos los posibles casos de usos de la aplicación, detallaremos la navegación así como la estructura de la base de datos y finalmente veremos el ciclo de vida de un archivo todo ello en cuanto a la arquitectura de la aplicación.

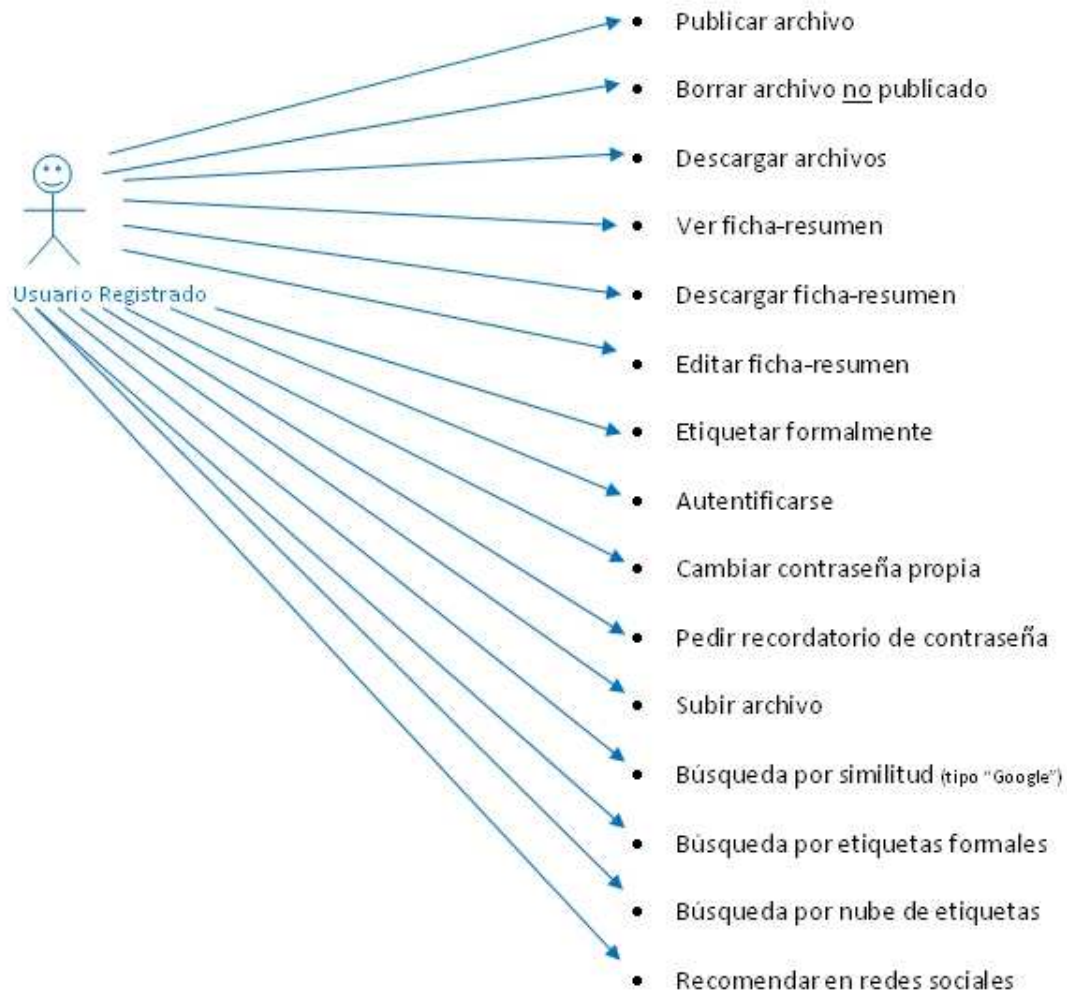
Más adelante describiremos la funcionalidad de la aplicación y finalmente concluiremos comentando las principales dificultades, conclusiones y trabajo futuro que se podría realizar a partir del actual repositorio RepASA.



# Capítulo 5: Arquitectura de la aplicación

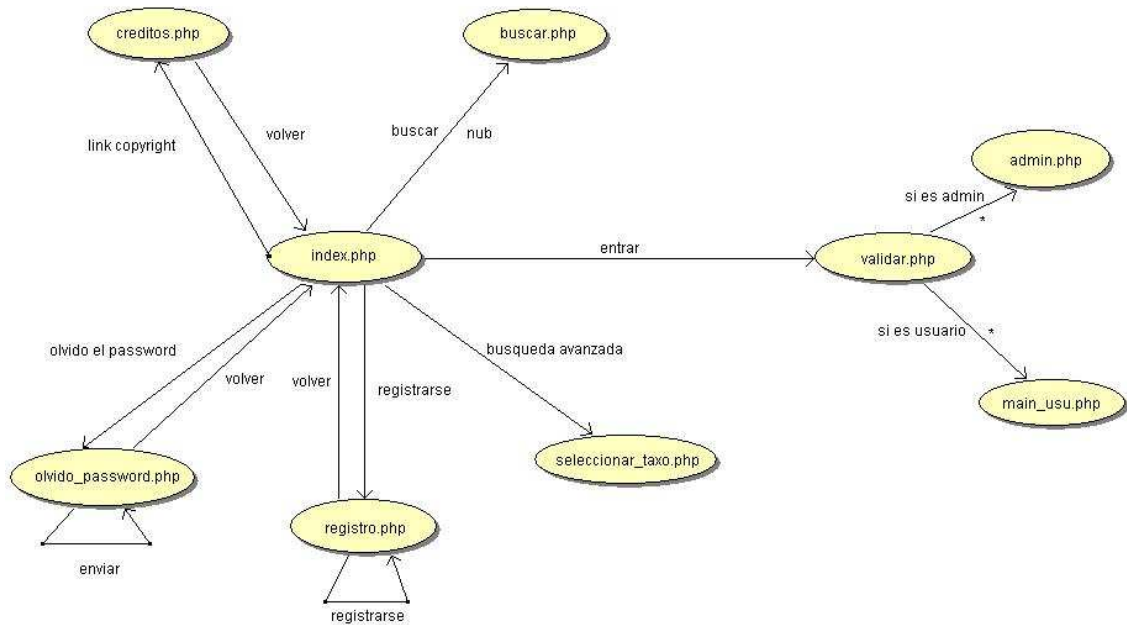
## 5.1 Casos de uso



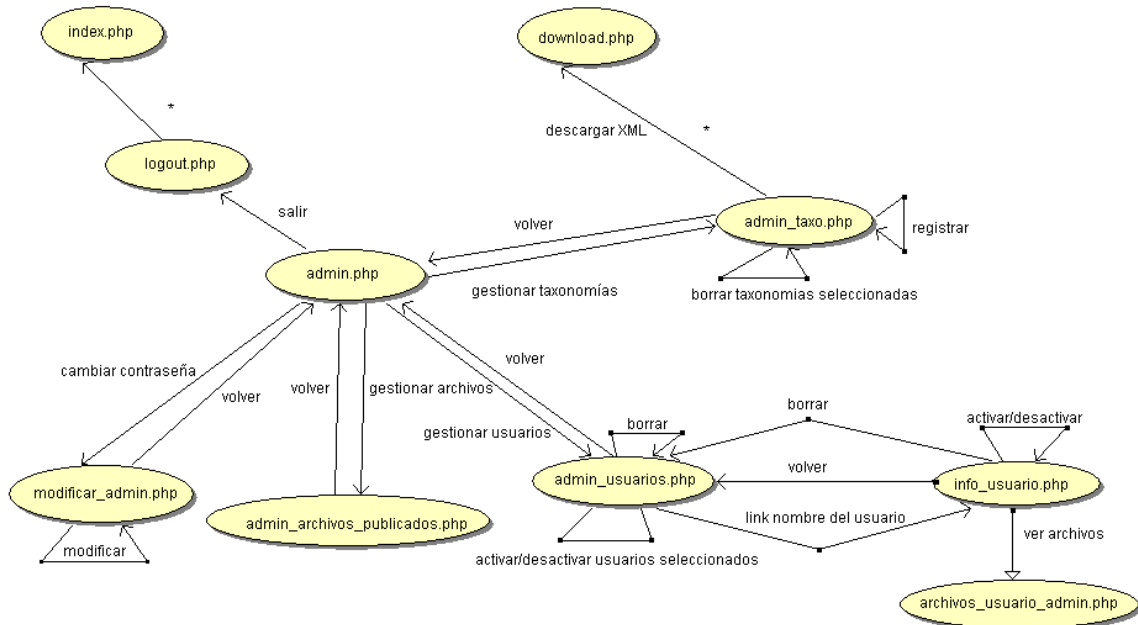


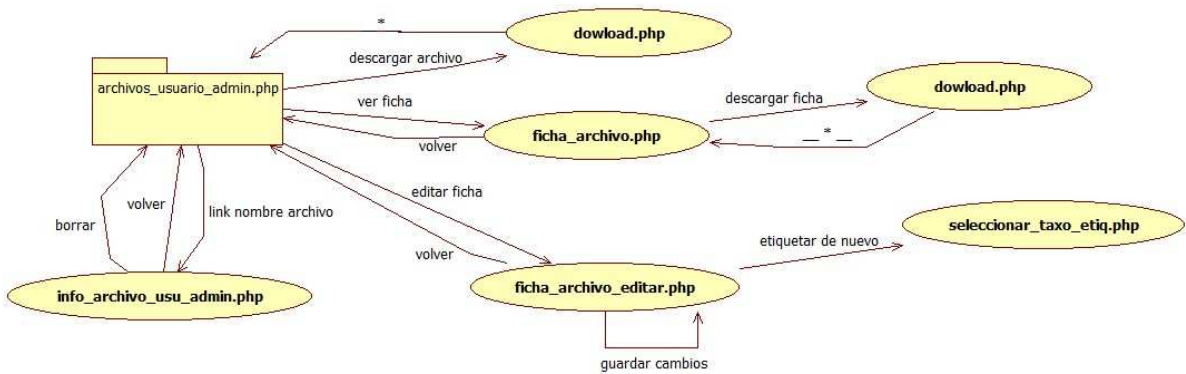
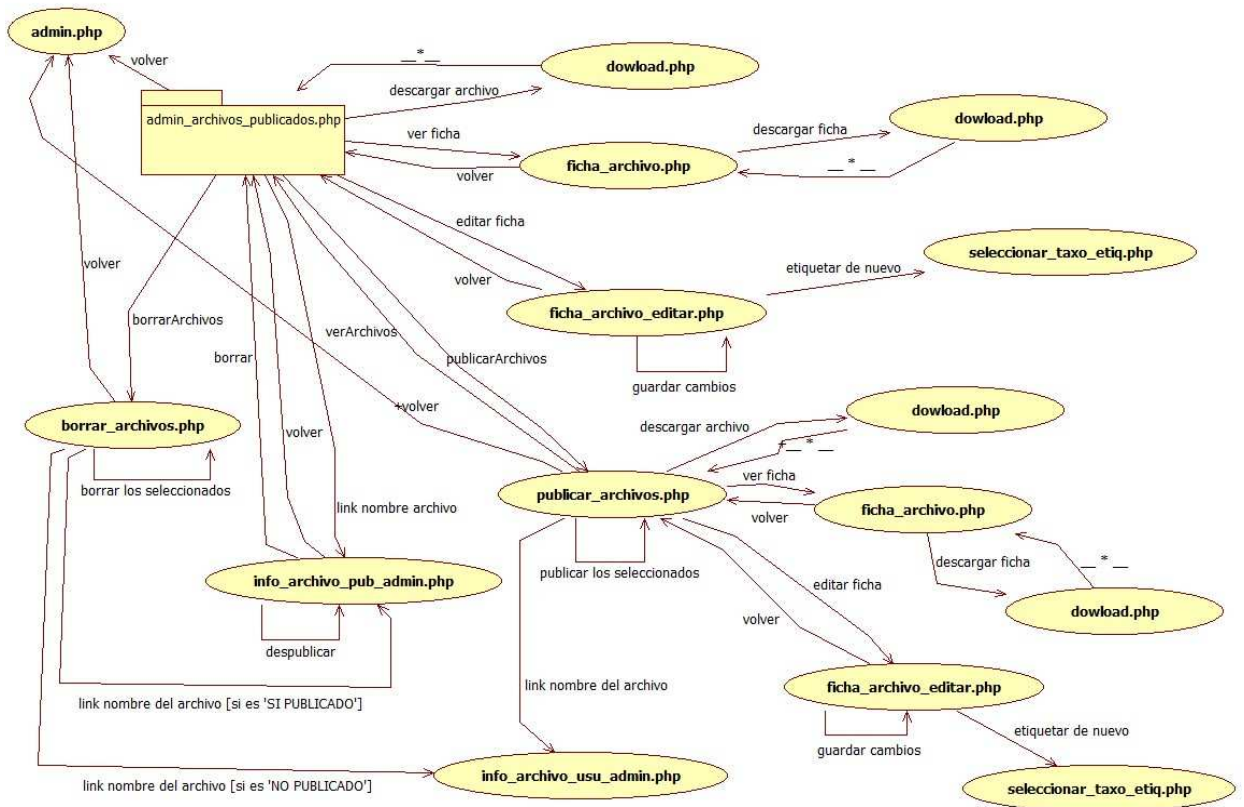
## 5.2 Navegación por las páginas PHP

### 5.2.1 Usuario no registrado (página Index)

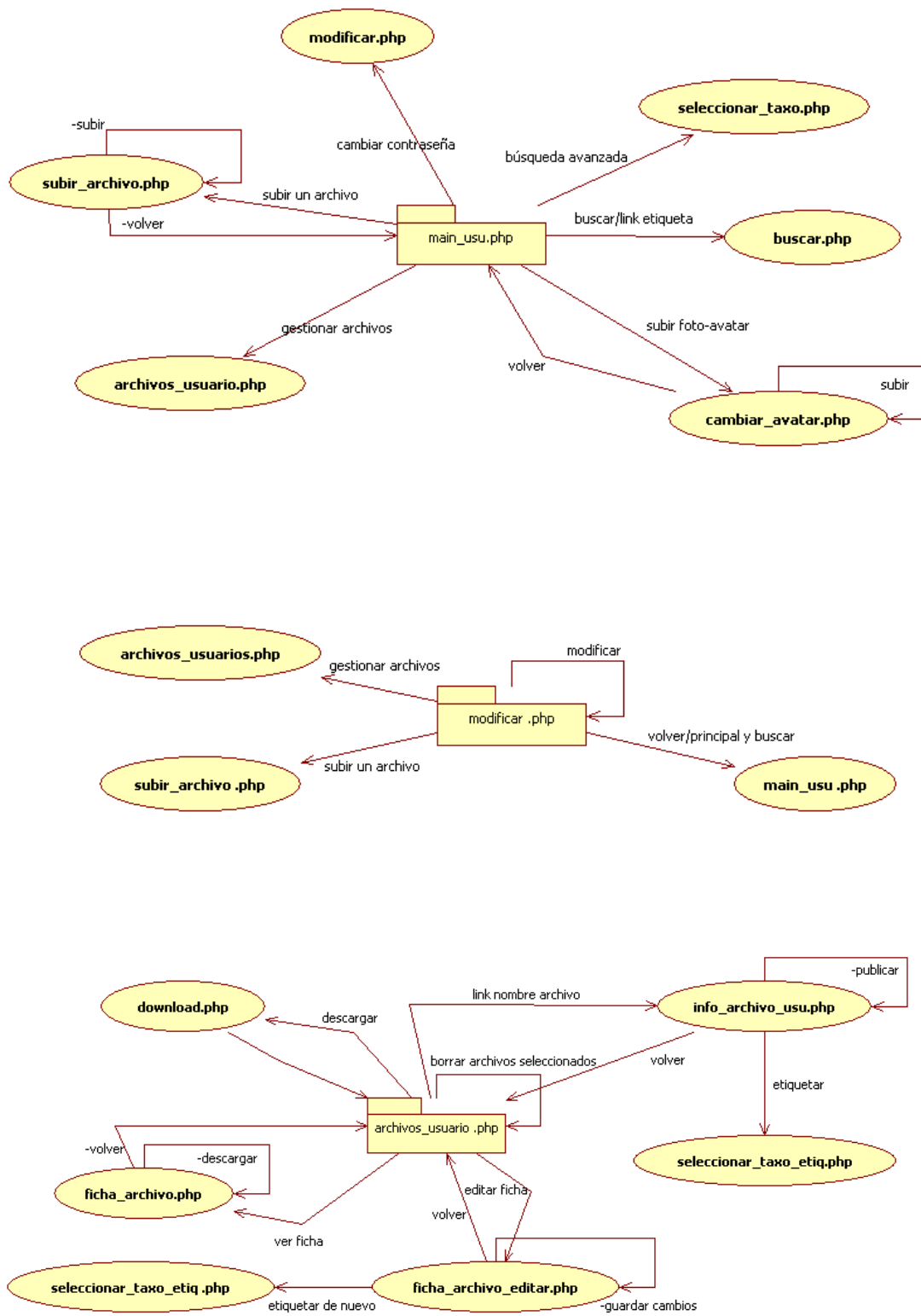


### 5.2.2 Administrador





### 5.2.3 Usuario registrado



### 5.3 Inclusiones entre archivos PHP

busqueda\_g ∈ búsquedas  
busqueda\_t ∈ búsquedas

busquedas ∈ index  
busquedas ∈ main\_usu

nube ∈ index  
nube ∈ main\_usu

estilo\_nube ∈ index  
estilo\_nube\_peq ∈ main\_usu

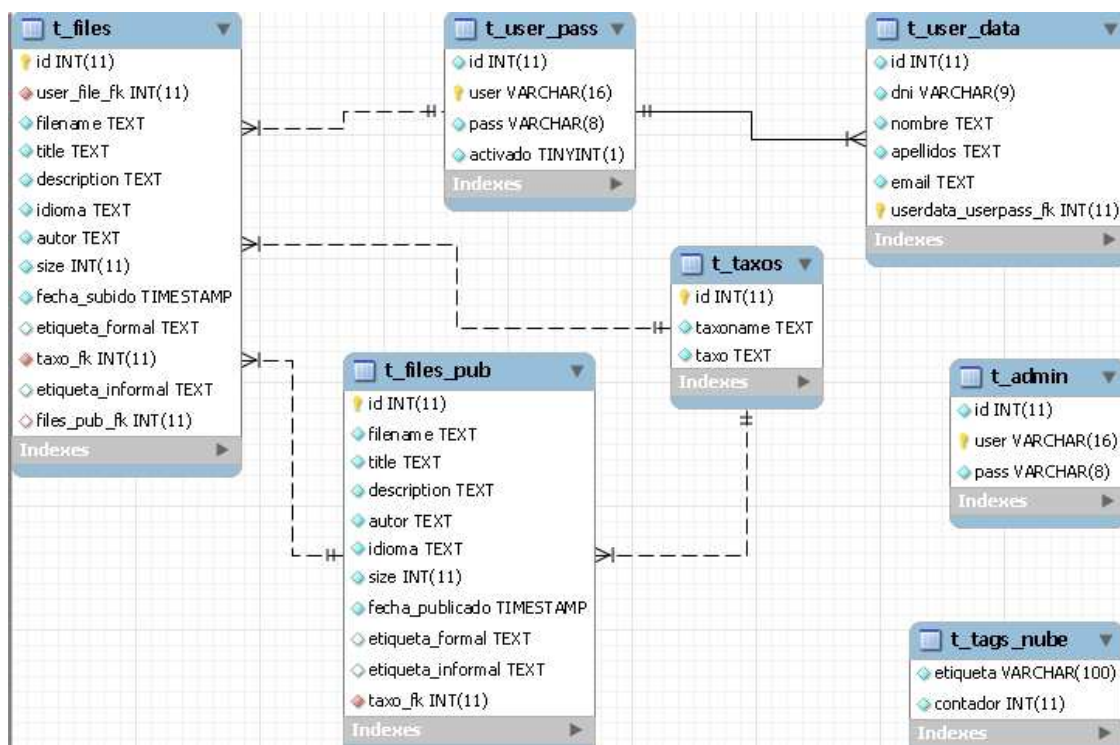
estilo\_repo ∈ todos los archivos la incluyen  
estilo\_ie ∈ todos los archivos la incluyen

class.smtp ∈ olvido\_password  
class.phpmailes ∈ olvido\_password

### 5.4 Estructura de la base de datos

La base de datos que se ha diseñado para el almacenamiento de toda la información necesaria de nuestro repositorio RepASA cuenta con la siguiente estructura, que explicaremos a continuación.

Tenemos un total de 7 tablas que forman la base de datos, las cuales están relacionadas de la siguiente forma, tal y como muestra el siguiente diagrama de entidad relación:



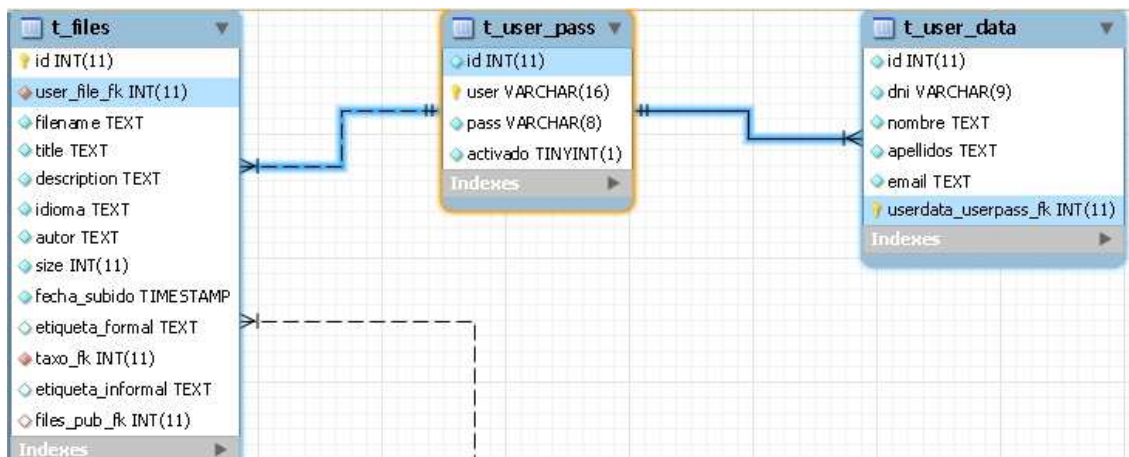
Vamos ir viendo cada una de las tablas para explicar su contenido y sus relaciones:

## 1. - t\_admin



Almacena el usuario y la contraseña del administrador.

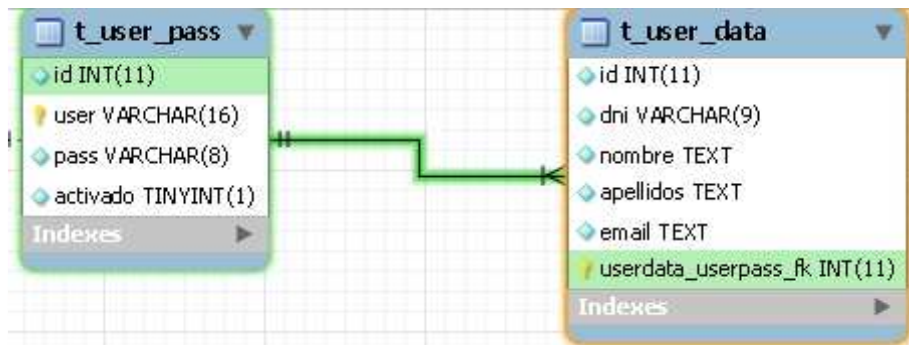
## 2. - t\_user\_pass



Almacena los nombres de usuarios y las contraseñas de todos los usuarios registrados en el sistema.

También tiene el campo 'activado', el cual guarda el entero que indica si el usuario está activado (el campo vale 1 y el usuario puede acceder al repositorio) o desactivado (el campo vale 0 y el usuario no puede acceder al repositorio hasta que el administrador no le active).

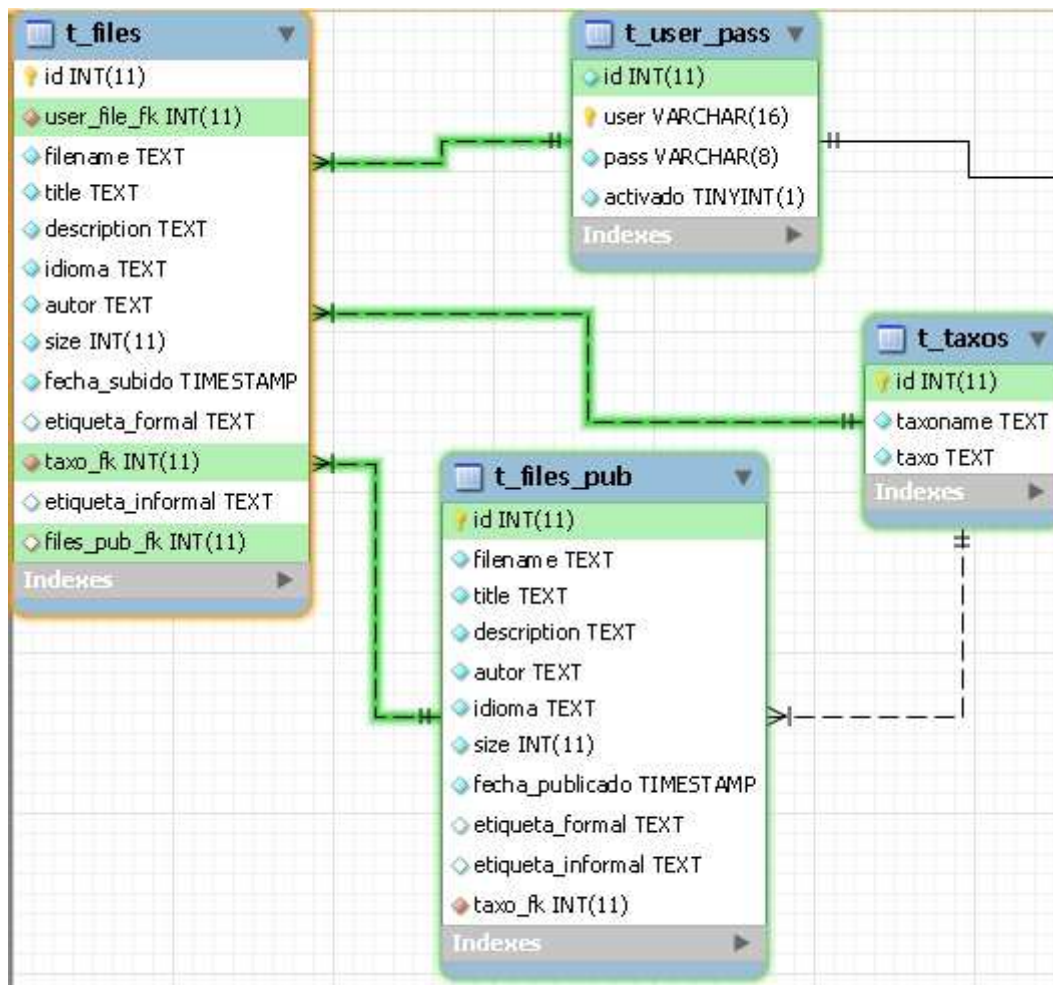
### 3. - t\_user\_data



Almacena los datos del usuario: nombre, apellidos, email, DNI.

El campo `userdata_userpass_fk` relaciona esta tabla con la tabla `t_user_pass`.

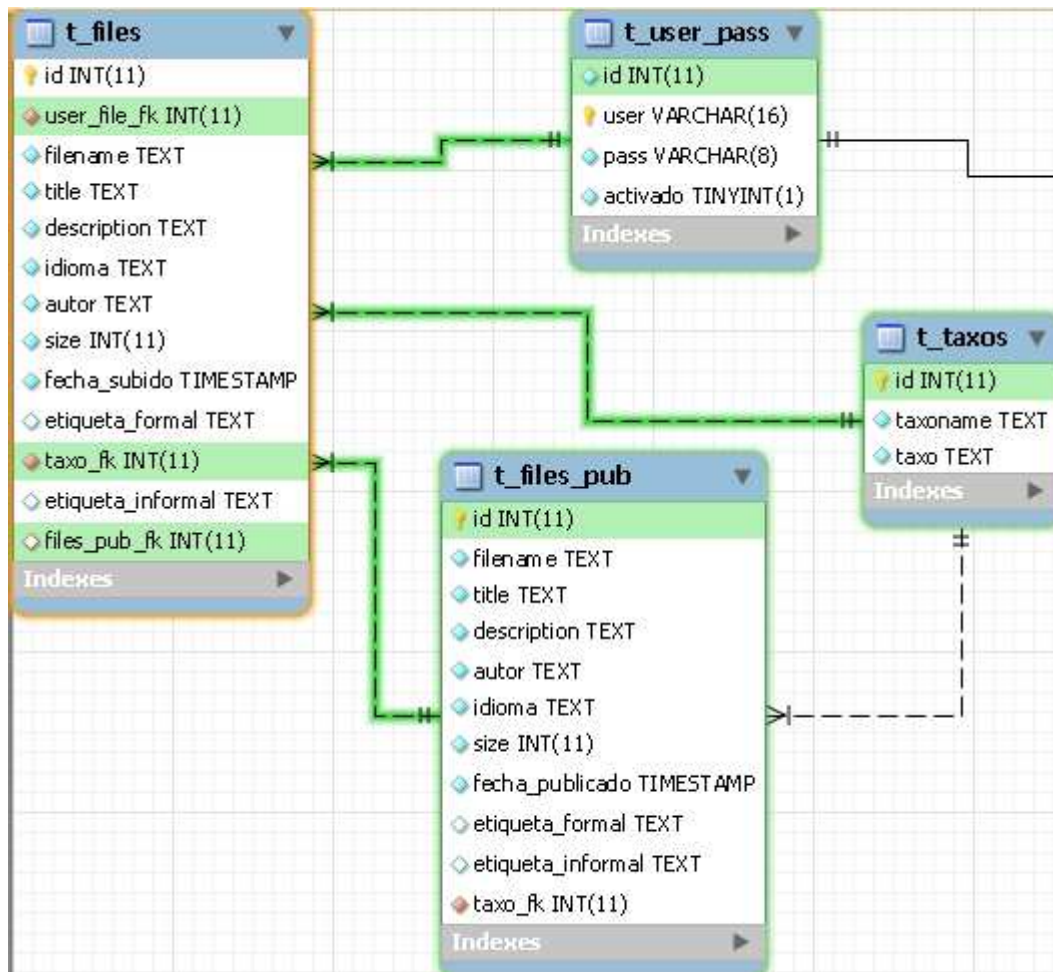
#### 4. - t\_files



Almacena los datos de los archivos que suben los usuarios al repositorio: nombre del fichero, título, descripción, idioma, autor, tamaño, fecha en la que se ha subido, palabras clave (etiqueta informal), la etiqueta formal obtenida de la taxonomía con la que se ha etiquetado, la propia taxonomía (taxo\_fk, que relaciona esta tabla con la tabla t\_taxos) y el usuario que lo ha subido (user\_file\_fk, que relaciona esta tabla con la tabla t\_user\_pass).

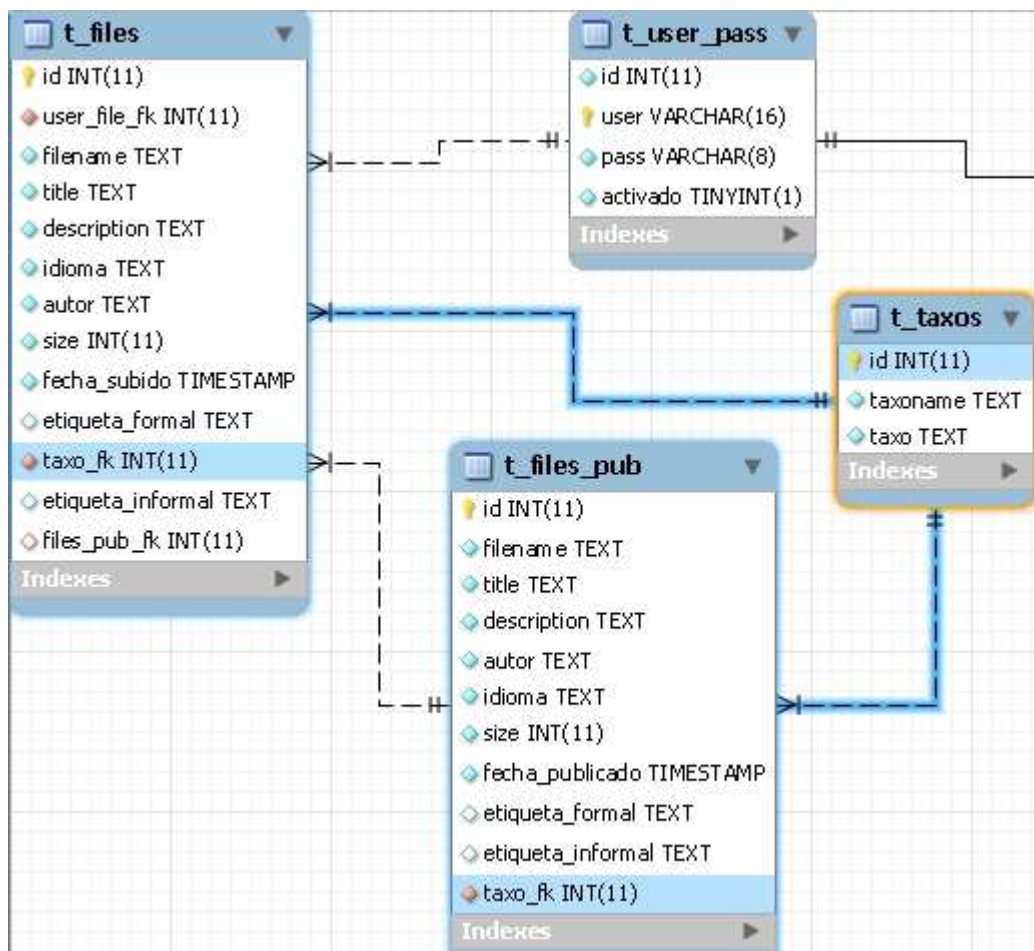
También, en el caso de que el archivo esté publicado, el campo files\_pub\_fk relaciona esta tabla con la tabla donde se almacenan los datos de los archivos publicados (t\_files\_pub).

## 5. - t\_files\_pub



Almacena los datos de los archivos que se han publicado en el repositorio: nombre del fichero, título, descripción, idioma, autor, tamaño, fecha en la que se ha publicado, palabras clave (etiqueta informal), la etiqueta formal obtenida de la taxonomía con la que se ha etiquetado y la propia taxonomía (taxo\_fk, que relaciona esta tabla con la tabla t\_taxos).

## 6. - t\_taxos



Almacena el nombre que da el administrador a la taxonomía y el fichero de la misma.

## 7. - t\_tags\_nube



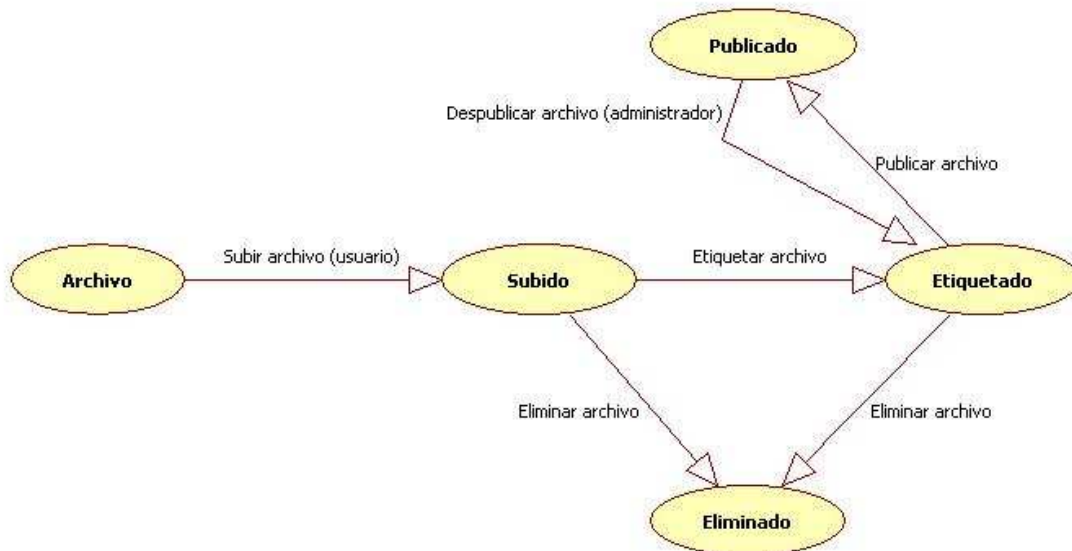
Almacena las palabras clave o etiquetas y el número de veces que se han buscado en el repositorio.

## 5.5 Ciclo de vida de un archivo

Una vez que un usuario registrado en el repositorio sube un archivo, éste puede pasar por los siguientes estados:

- Subido: el archivo ha sido subido por el usuario al repositorio.
- Etiquetado: el usuario o el administrador ha etiquetado el archivo.
- Publicado: el usuario o el administrador ha publicado el archivo. Sólo el administrador podrá despublicarlo.
- Eliminado: el usuario o el administrador ha eliminado el archivo.

El siguiente diagrama muestra el ciclo de vida del un archivo:



## Capítulo 6: Funcionalidad de la aplicación

Como hemos visto, en la actualidad existen muchos repositorios. Cuando este proyecto todavía estaba ideándose, teníamos ante nosotros la posibilidad de estudiar uno ya existente y dotarlo de mejoras, o crear desde cero uno nuevo.

Nos decantamos por la última opción. Por lo tanto, desde que RepASA empezó a cobrar forma, pensamos en todo momento en que nuestro proyecto sería la base de otros en el futuro. Y en eso centramos nuestros esfuerzos: no queríamos numerosas pequeñas funcionalidades a medias, sino unos cimientos sólidos que cualquiera pudiera recuperar, entender, reutilizar y ampliar.

Este repositorio permite buscar y recuperar objetos mediante metadatos que los usuarios registrados editan acerca de ellos (etiquetas formales escogidas de entre las categorías de una taxonomía dada y etiquetas libres).

Nos da garantías de seguridad obligando a los usuarios registrados a acceder introduciendo usuario y contraseña que se contrastan en una base de datos y manteniendo sesiones activas mientras estos acceden; de forma que, no pueda colarse ningún intruso no registrado ni mostrarse a un usuario el área personal de otro. Además, por si un usuario olvida su password, existe un sistema recordatorio de contraseña que únicamente se la reenviará si, además del "Usuario", introduce un email y éste es el que corresponde a ese usuario. (De esta forma evitamos que alguien hiciera un uso deshonesto del recordatorio de contraseña introduciendo nombres de usuario al azar o que conozca). Ese email con la contraseña les llegará automáticamente desde una dirección de correo creada expresamente para RepASA: [no.reply.repasa@gmail.com](mailto:no.reply.repasa@gmail.com)

Los usuarios registrados pero también los no registrados pueden buscar de tres formas distintas los archivos: búsqueda por similitud o 'tipo Google', búsqueda por etiquetas formales y búsqueda a través de una nube de etiquetas que se genera y aumenta con cada nueva búsqueda sus porcentajes de aparición de las etiquetas más buscadas.

Todos los usuarios pueden ojear todos los archivos publicados (un usuario puede subir un archivo pero no querer publicarlo todavía) y, cuando encuentren uno que les interesa, ver y, si lo desean, descargar, bien el archivo, o bien una ficha resumen del mismo.


El administrador, que accede también mediante un usuario y contraseña que se guardan en una base de datos diferente para mayor seguridad en caso de fallo (y mayor eficiencia en el acceso, pues busca sólo entre los administradores), puede gestionar usuarios (activarles, desactivarles, eliminarles por completo si fuera necesario...), los archivos de estos (publicar\*, despublicar, eliminarlos...), gestionar las taxonomías sobre las que se etiquetarán los objetos...

\* Que el administrador pueda hacer público un archivo no publicado de un usuario, sirve para que no se utilice el repositorio como un disco duro virtual; ya que, lo que se suba 'corre el peligro' de ser publicado. Todas estas funcionalidades de cada tipo de usuario (administrador, usuario registrado y usuario no registrado) se explican más profundamente y paso a paso en las secciones 6.3, 6.4 y 6.5.

Así que, queda por comentar los detalles del repositorio en sí mismo, sin tener en cuenta las funcionalidades dirigidas a los usuarios o al administrador.

Para RepASA, además del nombre, creamos un logotipo inventado y diseñado por nosotros, el cual, teniendo los colores corporativos del resto del repositorio, con verlo trajera a la cabeza la idea de que efectivamente te encuentras ante un lugar donde se almacena saber y conocimiento, libres.



Como pequeño icono que acompaña al nombre en las pestañas del navegador ("fav-icon"), elegimos la misma imagen que aparece en el avatar por defecto cuando el usuario no sube una propia personalizada: 

También pensando en los navegadores, añadimos un par de funcionalidades destinadas a evitar posibles incompatibilidades debidas a las diferentes interpretaciones y estándares que siguen unos y otros.

Se añadió en las páginas intermedias que, por comodidad, tienen redirecciones automáticas (como "validar.php" o "logout.php"), un enlace permanente con el mensaje: "Si tu navegador no te redirige en unos segundos, haz clic aquí".

Y para asegurar la compatibilidad con Internet Explorer (IE), el navegador de Microsoft y que más problemas nos dio, tuvimos que añadir en muchos sitios (pestañas del panel de usuario registrado, las nubes de etiquetas...) 'parches' con trozos de código que solamente IE leyera y ejecutara\*, e incluso crear un archivo CSS especial con atributos pensados para cuando IE descolocara algunos elementos.

\* IE interpreta como código...:

```
<!-- [if IE] Aquí pones el código [end if] -->
```

\* IE cree que son comentarios e ignora el código (que los demás navegadores sí leen) que haya entre los tags...:

```
<comment> </comment>
```

Las secciones siguientes secciones, 6.1 y 6.2, recalcan lo que ya comentamos de que este proyecto pretende ser los cimientos de otros posibles proyectos más avanzados y concretos. Hablan del esmero que pusimos en que fuera general, fácilmente entendible y modificable, y también, accesible para cualquier tipo de usuario.

## 6.1 Generalidad

En todo momento durante el transcurso del proyecto, se pensó en la necesidad de que fuera lo más general posible de cara a ser adaptable en un futuro a lo que se quiera implementar sobre él.

Programar código reutilizable y que no fuera, como sucede a veces, exclusivo para nuestro proyecto. Si no se tiene presente esto, a veces se podría acabar programando de forma que, cuando se cambie un detalle, ya no sirva el código tal cual está...

La parte gráfica del repositorio, sus colores representativos, tipos de letra y colores de ésta, son fácilmente modificables gracias a la utilización de la hoja de estilos CSS en la que están todos esos atributos claramente identificados por comentarios explicativos.

Para elegir ese aspecto y esos colores nos basamos en la plantilla "Neonix" de Free CSS Templates [4] y después, la cambiamos casi totalmente con Adobe Photoshop para que ya no tuviera un tamaño fijo conseguido con una única imagen de fondo, sino que se adaptase a cualquier tamaño de ventana. Esto lo logramos haciendo que el característico recuadro blanco que delimita el área de RepASA estuviera formado por 4 imágenes de esquinas e imágenes de 1px de ancho unas horizontales y otras verticales que se repiten hasta alcanzar el tamaño necesario.

También atributos que no se refieren a los colores pero sí al a forma o colocación de algunos elementos que en un principio estaban en el código, pasaron a especificarse desde la hoja de estilos.

Por ejemplo el ancho de la tabla general común a todas las páginas, que pasó de estar en todas las páginas como `<table align="center" width="60%">` (con lo que, para cambiar ese ancho habría que abrir muchos archivos) a tener en todos la línea: `<table class="anchotabla">` (con la cual, si se quiere cambiar ese tamaño, con modificar la hoja de estilos se cambiará automáticamente en todas las páginas a la vez)

Las imágenes e iconos que se usan para dar vistosidad en listas de usuarios y archivos (luces roja/verde/naranja, iconos de la ficha-resumen...), todas son gráficos PNG con fondo transparente. De esa forma, si se quiere cambiar los colores de fondo, no será necesario tocar las imágenes.

Las imágenes que forman la línea blanca que delimita el área del repositorio, en un principio tenían blanco y verde pero, posteriormente, también pasaron a ser PNG con fondo transparente. De esa forma, al igual que antes, puede cambiarse el fondo sin que el cambio repercuta en nada más.

Todas las rutas entre páginas son relativas. No hay rutas absolutas. De esa forma funcionará tanto si se ejecuta en la propia máquina de uno en un servidor local como si se sube a internet y se le confiere un dominio real desde el que acceder.

## 6.2 Accesibilidad

La preocupación por la accesibilidad web es necesaria si tenemos en mente a la variedad de personas que pueden querer hacer uso de nuestro repositorio.

No se debería en ningún proyecto, pero menos aún en uno concebido para internet como es RepASA, dejar de lado los posibles casos de personas que, por sus circunstancias (discapacidades, monitores antiguos con resoluciones de pantalla anormalmente pequeñas, conexiones lentas...), no utilicen sus ordenadores y equipos informáticos de la misma forma que el común de demás usuarios.

RepAsa ha sido creado, desde sus inicios, teniendo claro ese punto: es un repositorio básico accesible para todos.

Por eso, en nuestro código hemos trabajado observando en todo momento todas las pautas recomendadas para que un sitio web se considere accesible: [ver Anexo “Normativas Accesibilidad Web”]

Como hemos mencionado en el punto anterior, la parte estética del repositorio está toda en un archivo CSS u hoja de estilos que se encarga de fijar los colores e imágenes de fondo, las pestañas, los logotipos de todas las cabeceras y de fijar (mediante atributos del tag html “<div>”) la posición relativa de los diferentes elementos que componen cada página. De esta forma, gracias a estar separados el contenido del repositorio y la parte que es únicamente estilo, la navegación de los posibles usuarios con problemas de algún tipo, es más fácil y fluida.

Además, sin esta página de estilo se sigue visualizando bien y tiene sentido completo. Lo cual es, también, un requisito necesario para la accesibilidad web ya que algunos intérpretes para discapacitados visuales ignoran estas hojas ya que, en principio, no aportan nada de información a sus usuarios\*.

\* La parte de estilo no aporta nada nuevo a los usuarios siempre y cuando, al programar, se cumpla que toda la información que reciba el usuario mediante imágenes o colores (por ejemplo rojo=malo, verde=bueno), esté presente también en el código... (Como diremos más abajo RepASA cumple este requisito).

En la siguiente imagen podemos ver cómo queda RepASA si no se le aplica la hoja de estilo: los elementos dejan de estar situados en donde ‘deberían’ y dejan de adoptar las formas originales (como es el caso de la nube de etiquetas); pero todo sigue teniendo sentido completo y se visualiza perfectamente.

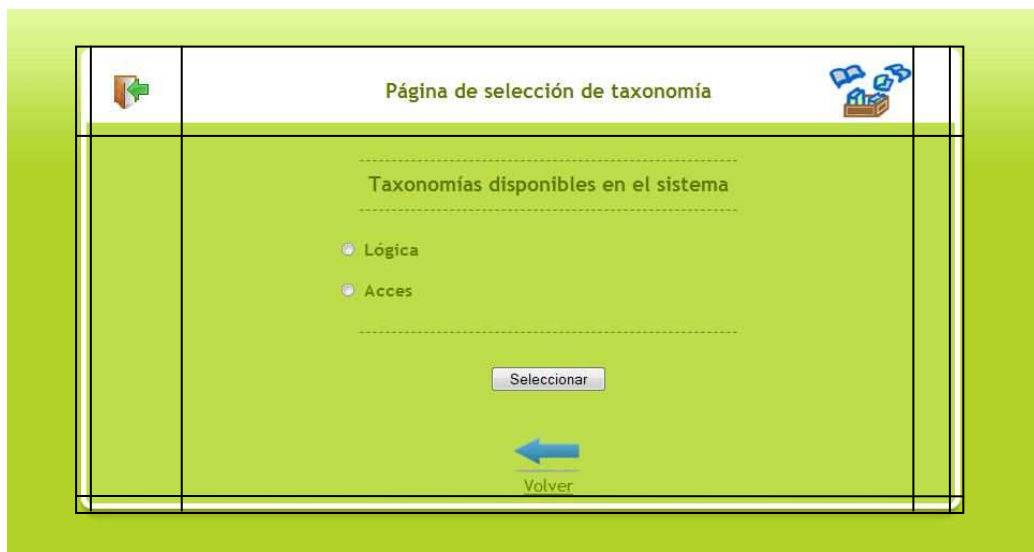


Otra pauta es que “se recomienda no usar tablas para maquetar y colocar los elementos”. (Es decir: no usar, por ejemplo, celdas en blanco para generar espacios). Y añade: “a menos que la tabla tenga sentido cuando se alinee.”

Como vemos en la imagen anterior, el elemento “acceder” mantiene su estructura; esto es porque se trata de una tabla de 3 filas y 2 columnas. Pero esta tabla, cuando un intérprete la alinee, sigue teniendo sentido completo. La tabla transformada en una línea quedaría de la siguiente manera: “Usuario”, campo donde escribir, “Password”, campo donde escribir, botón “Entrar”, link “¿olvidó el password?”, link “Registrarse”.

Algo parecido ocurre con la tabla general que rodea a la totalidad del código de cada página:

La tabla, idéntica para todas las páginas, tiene esta estructura:



Que, una vez alineada quedaría:

Imagen esquina, link “salir”, título-cabecera, celda\_vacía\*, imagen esquina, borde izquierdo, celda\_vacía\*\*, CONTENIDO, celda\_vacía\*\*, borde derecho, imagen esquina, celda\_vacía\*\*, borde abajo, celda\_vacía\*\*, imagen esquina.

\* La celda vacía marcada con (\*) está para proporcionar el ancho a las demás de su columna, es decir, esa celda está siendo usada para el formateo y no cumple con el requisito de accesibilidad.



\*\* Las celdas (\*\*) no entorpecen la lectura accesible, pues están precisamente para mantener una estructura homogénea de tabla cuadrada evitando la combinación de celdas, que la harían menos comprensible.

Se optó por una tabla a pesar de perder en accesibilidad, para poder ganar en generalidad, ya que, como hemos explicado en el punto anterior, es gracias a ello que RepASA puede adaptarse y verse con el mismo aspecto en cualquier resolución de pantalla.

Todos los demás elementos están posicionados mediante atributos del CSS.

Todas las imágenes e hipervínculos tienen siempre asociados su texto alternativo (atributo “alt”) para cuando no se carguen correctamente, y su descripción (atributo “title”), que se muestra al pasar el cursor del ratón por encima de ellos.

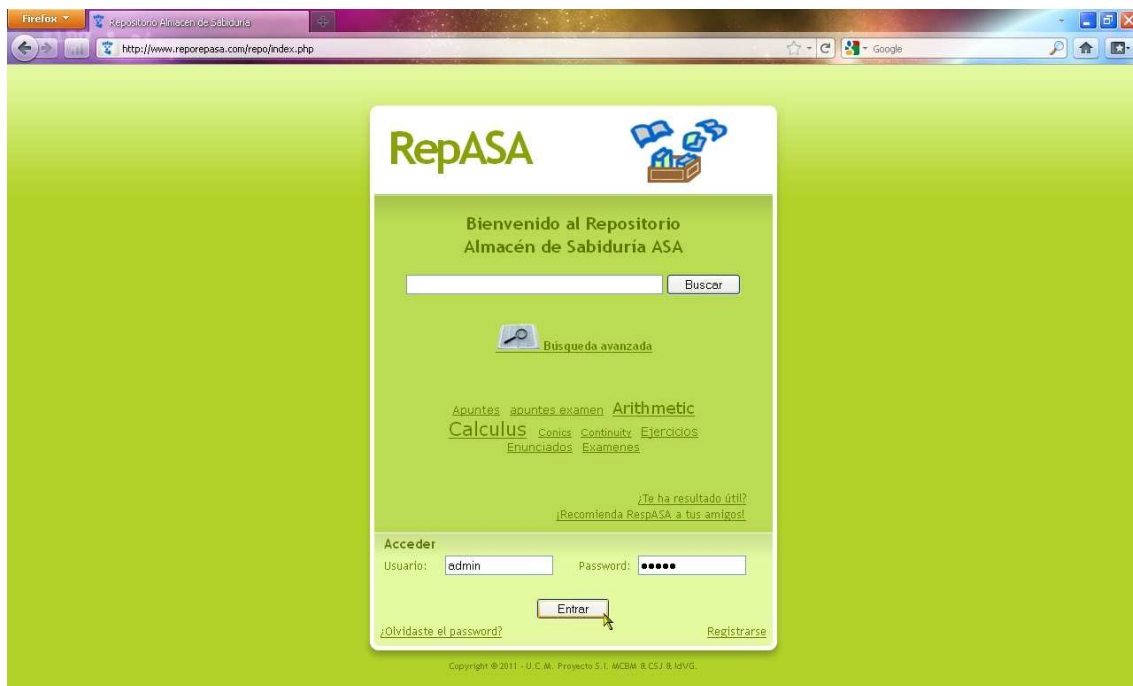
**De esta forma, cualquier información que se le da al usuario mediante colores, le llega igualmente aun cuando no tenga la posibilidad de visualizar imágenes.**

Por poner un ejemplo...: Las imágenes “luz roja”  (archivo no publicado / usuario no activado) y “luz verde”  (publicado/activado) tienen como texto alternativo a ellas “(-)” y “(+)” respectivamente.

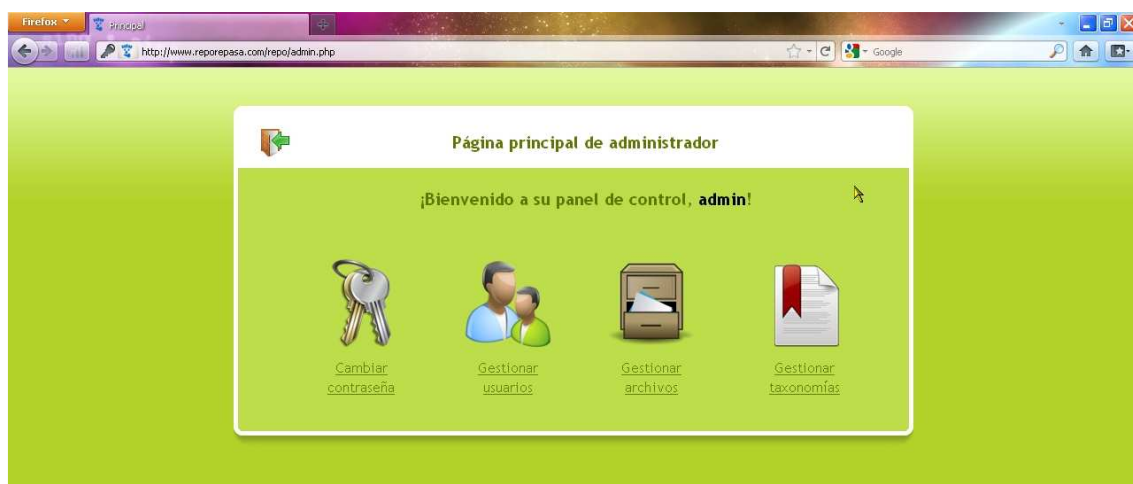
### 6.3 Manual de explotación del administrador

El administrador del sistema tiene disponibles una serie de funcionalidades que le va a permitir una fácil e intuitiva gestión de todo el repositorio, desde la gestión de usuarios y archivos, hasta la posibilidad de incorporar nuevas taxonomías o cambiar su propia contraseña. Toda esta funcionalidad se explica paso a paso más adelante.

Primeramente, el usuario con el rol de administrador tiene que entrar en la página principal de RepASA y entrar con el usuario **admin** y la contraseña **admin**:



Una vez dentro de la aplicación, tenemos disponibles las siguientes funciones, las cuales se explicarán detalladamente:



## Cambiar Contraseña

Es posible modificar la contraseña del administrador, sólo basta con introducir una nueva (por duplicado) y pulsar Aceptar:



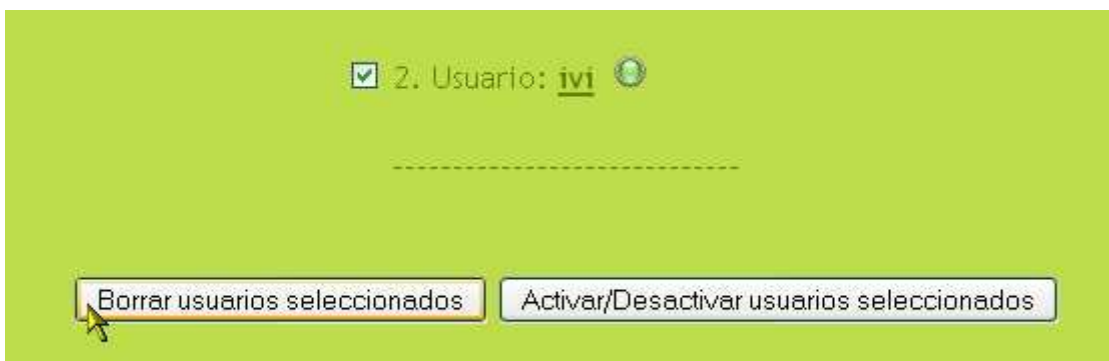
## Gestionar usuarios

Desde el menú de gestión de usuarios, se tiene acceso a todos los usuarios que están registrados en el sistema:



Desde esta página se pueden realizar las siguientes tareas:

- Borrar usuarios: para ello, se marcarán las casillas correspondientes a los usuarios que se desea eliminar y se pulsa el botón “Borrar usuarios seleccionados”:

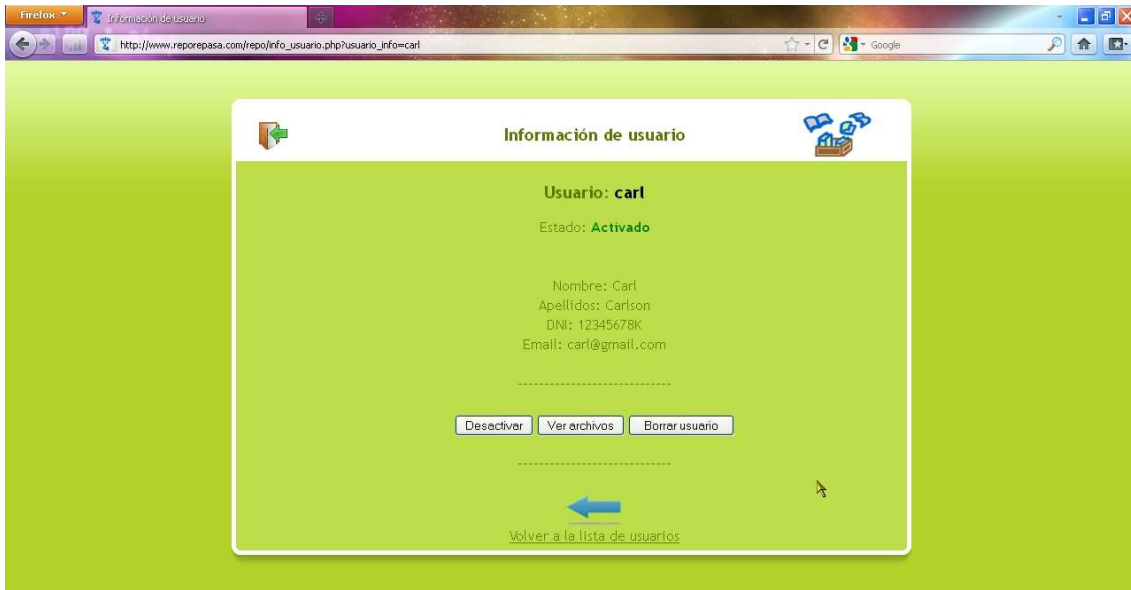


- Activar/Desactivar usuarios: para ello, se marcarán las casillas correspondientes a los usuarios que se quieran activar/desactivar y se pulsa el botón “Activar/Desactivar usuarios seleccionados”, de tal modo que se activarán los usuarios marcados y que estaban desactivados y se desactivarán los usuarios marcados que estaban activados.  
Una desactivación sobre un usuario impide que éste pueda acceder al sistema hasta que no se le vuelva a activar por parte del administrador.



Además, desde la página de gestión de usuarios, se puede acceder a la información de un usuario registrado pulsando el link del nombre de usuario:





Desde esta página, con la información del usuario, podemos borrar al usuario (pulsando el botón “Borrar usuario”), activar (desactivar) al usuario (pulsando el botón “Activar” (“Desactivar”) según en el estado que esté el usuario aparecerá un botón u otro) y ver los archivos que tiene el usuario (pulsando el botón “Ver archivos”), desde donde se podrá gestionarlos:



Desde aquí, podemos descargarnos el archivo:



Además podemos ver, descargar y modificar la ficha del archivo, pudiendo también volver a etiquetarlo (ver Gestionar archivos - Ver archivos):



Si pulsamos el link con el nombre del archivo, podremos etiquetarlo y publicarlo, si éste no lo estaba ya, además de poder borrarlo (ver Gestionar archivos - Publicar archivos):



## Gestionar archivos

En la gestión de archivos, tenemos las siguientes funcionalidades disponibles:

- **Ver archivos:**  
Desde esta pestaña podremos ver los archivos que han sido publicados en el repositorio, bien por el propio administrador, o bien por los usuarios registrados.



- Descargar archivo:  
Para descargar el archivo basta con pulsar el icono de descarga:



- Ver/descargar ficha:  
Si queremos ver la ficha pulsamos el icono:



Ahora podremos descargarla en un fichero de texto pulsando:



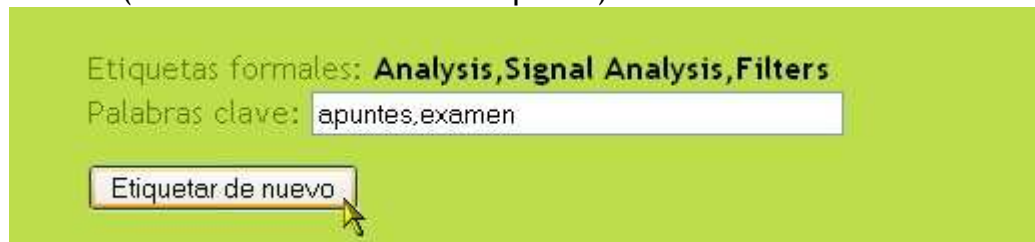
- Editar ficha:  
Para editar la información de la ficha del archivo, pulsaremos el icono de editar:



Podremos modificar todos los campos que nos aparezcan como modificables y cuando estén listos pulsaremos el botón “Guardar cambios”:



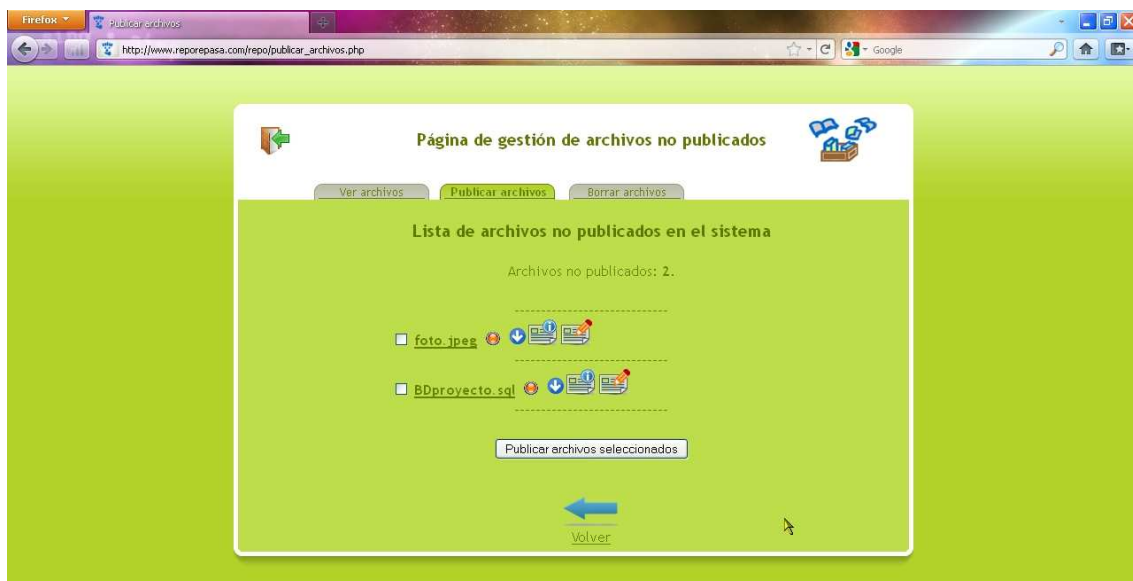
Para etiquetar de nuevo el archivo, pulsaremos “Etiquetar de nuevo” (ver Publicar archivos - Etiquetar):



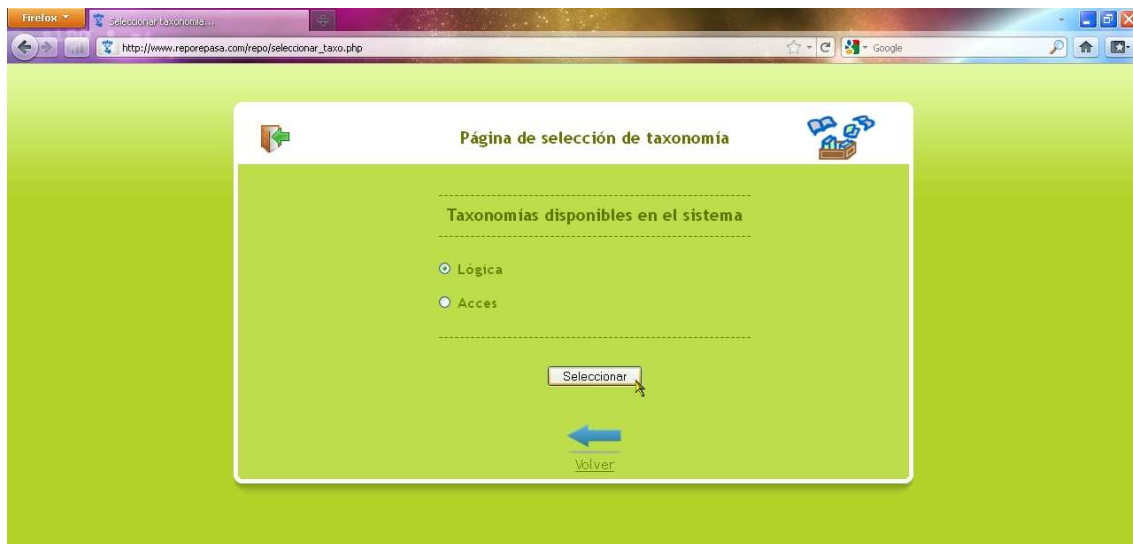
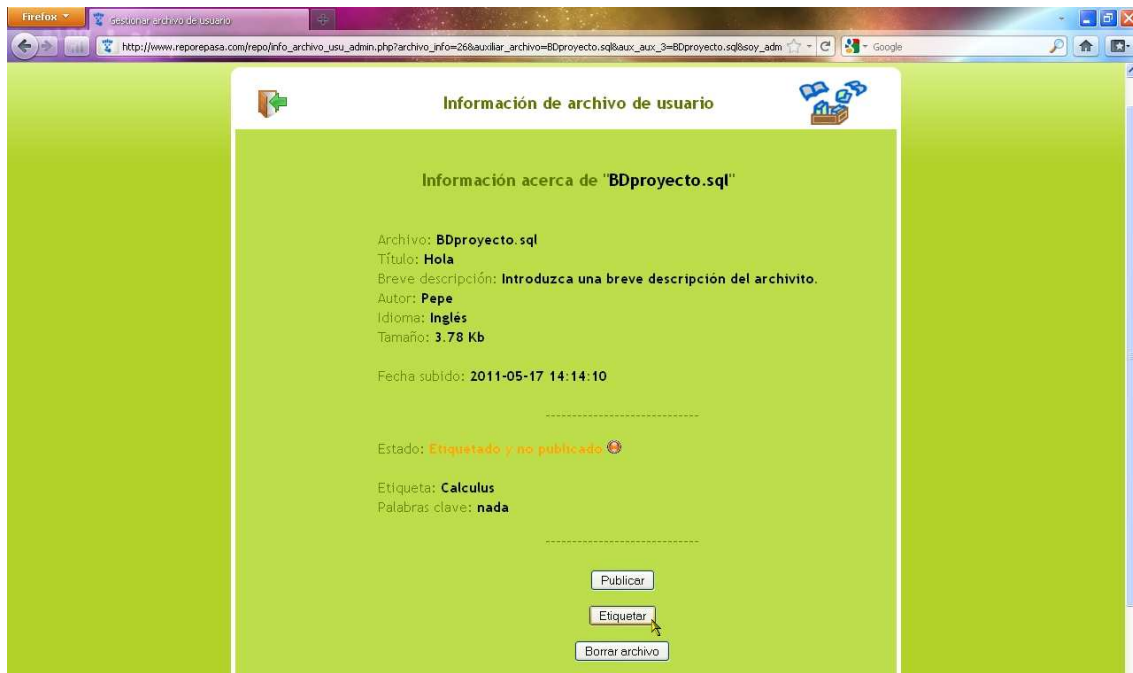
- Despublicar/Borrar archivo:  
Si queremos despublicar el archivo (los usuarios no podrán acceder a él, sólo el usuario que lo subió), o borrarlo, hacemos clic en el nombre del archivo y tendremos disponibles estas dos funciones:



- **Publicar archivos:**  
En esta pestaña el administrador puede publicar archivos que están etiquetados o bien etiquetarlos él antes de publicarlos.



Dentro de la información del archivo (pulsamos link del nombre) podremos etiquetar un archivo, para ello, pulsamos el botón “Etiquetar” que nos llevará a la página de selección de taxonomía. Seleccionamos la taxonomía con la que vamos a etiquetar y pulsamos el botón “Seleccionar taxonomía”:



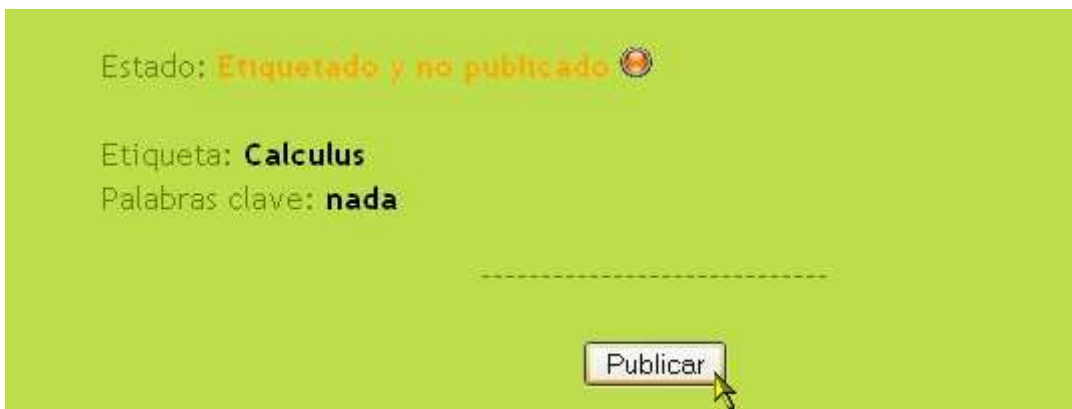
Ahora podemos ir navegando por las diferentes categorías de la taxonomía para ir construyendo la etiqueta formal. Cuando estemos dentro de la categoría con la que queremos marcar al archivo, pulsaremos el botón "Etiquetar":



Éste, nos llevará al siguiente paso del etiquetado, las palabras clave. Para realizar este paso, escribiremos las palabras clave que queremos etiquetar al archivo, separadas por comas, y pulsaremos el botón “Etiquetar”:



Ahora ya tenemos el archivo completamente etiquetado, por lo que ya podemos publicarlo pulsado el botón “Publicar”:



Ahora, el archivo queda a la vista de todos los usuarios que visiten el repositorio.

- **Borrar archivos:**

En esta última pestaña podremos eliminar archivos bien marcando las casillas correspondientes y pulsando el botón eliminar:



O bien, pulsando el link del nombre del archivo, entramos a la información del mismo y podemos borrarlo pulsando el botón "Borrar":



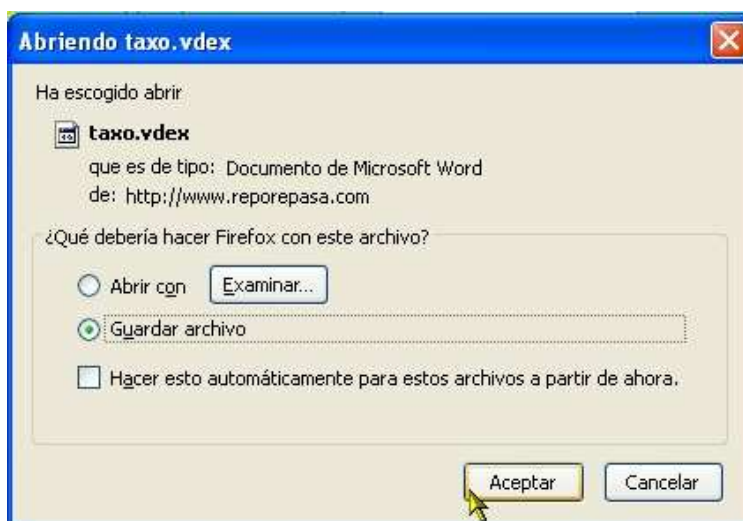
## Gestionar taxonomías

El administrador puede gestionar las taxonomías del sistema: dar de alta nuevas taxonomías o eliminar las que sean necesarias.

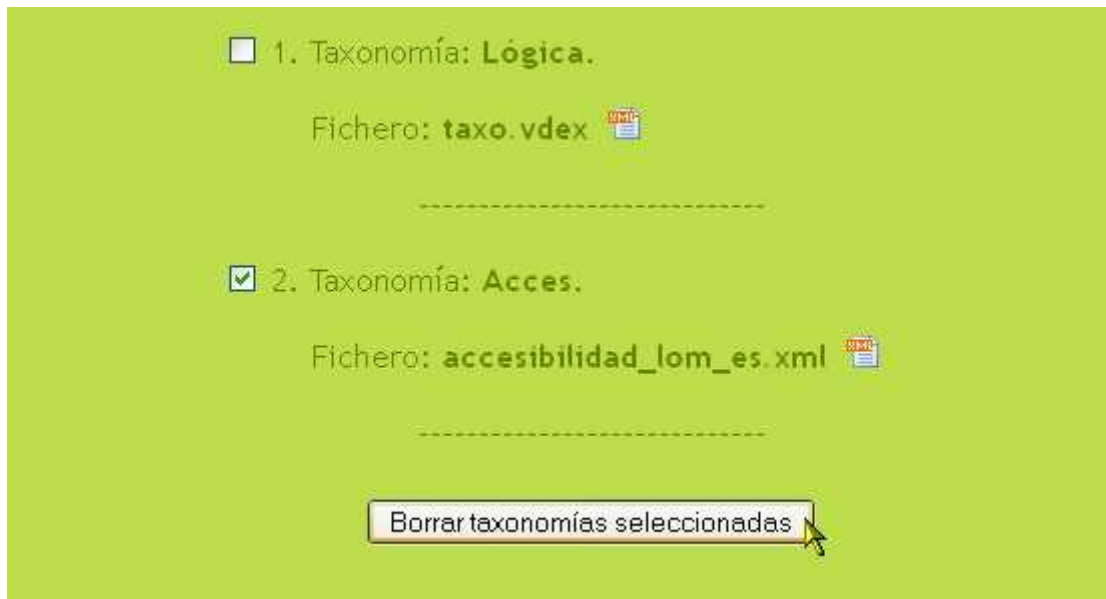
Para subir y dar de alta una nueva taxonomía se debe seleccionar el fichero XML que contenga dicha taxonomía, dar un nombre para identificarla en el sistema y pulsar el botón “Registrar”:



También se puede descargar una taxonomía previamente registrada en el sistema pulsando el icono que aparece junto al nombre de la taxonomía:



Para eliminar una taxonomía del sistema basta con seleccionarla y pulsar el botón 'Borrar taxonomías seleccionadas':



### Salir de la aplicación

Para salir de la aplicación basta con pulsar el icono de 'Salir' (esquina superior izquierda):

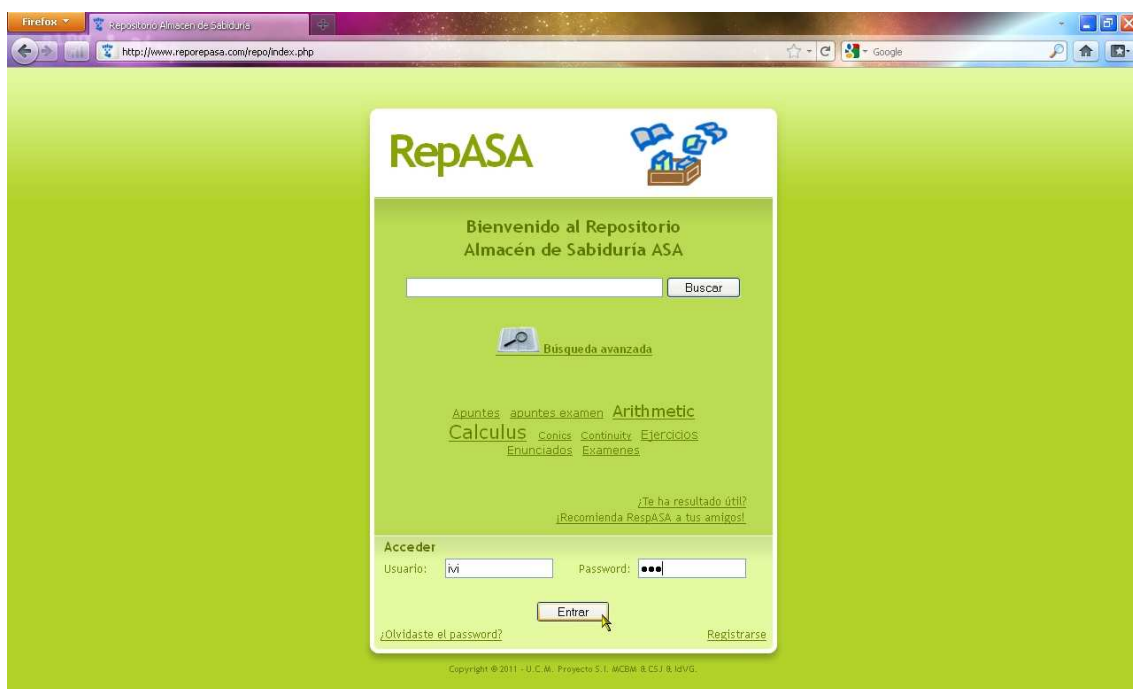


## 6.4 Manual de explotación del usuario registrado

El usuario registrado en el sistema tiene disponibles una serie de funcionalidades que le va a permitir una fácil e intuitiva gestión de sus archivos, la publicación de los mismos, añadir un avatar o cambiar su propia contraseña, teniendo en cuenta que además podrá buscar, descargar y editar las fichas de archivos que él no ha subido o publicado.

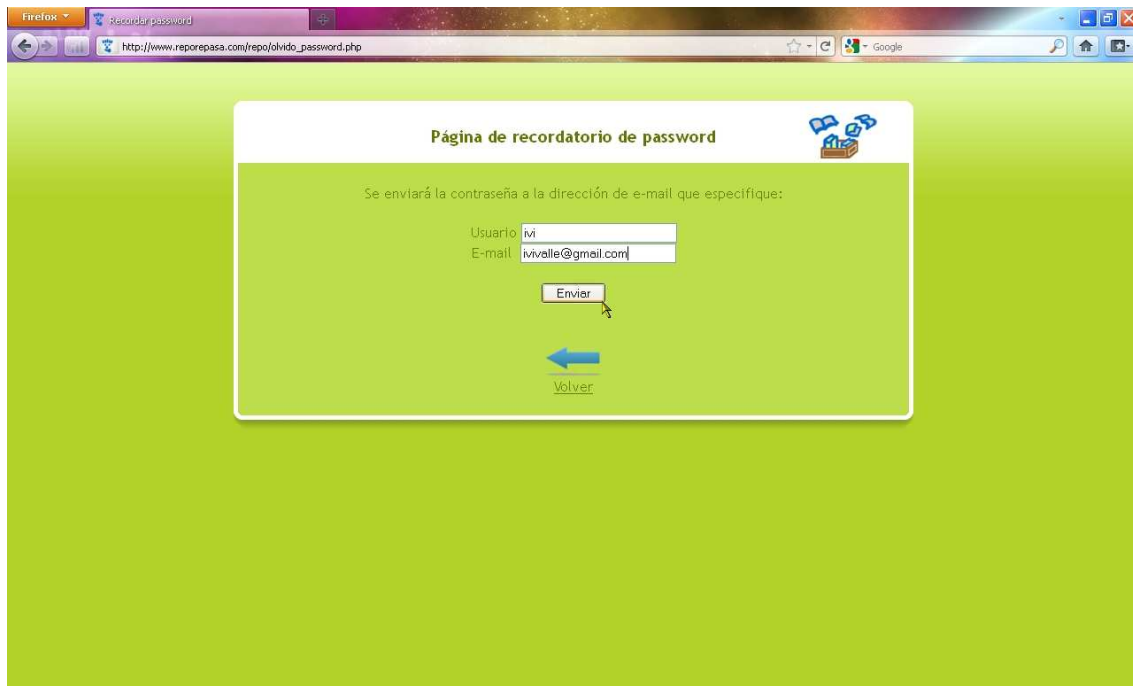
Toda la funcionalidad disponible se explica paso a paso a continuación:

Primeramente, el usuario tiene que entrar en la página principal de RepASA y entrar con su nombre de usuario y su contraseña\*:

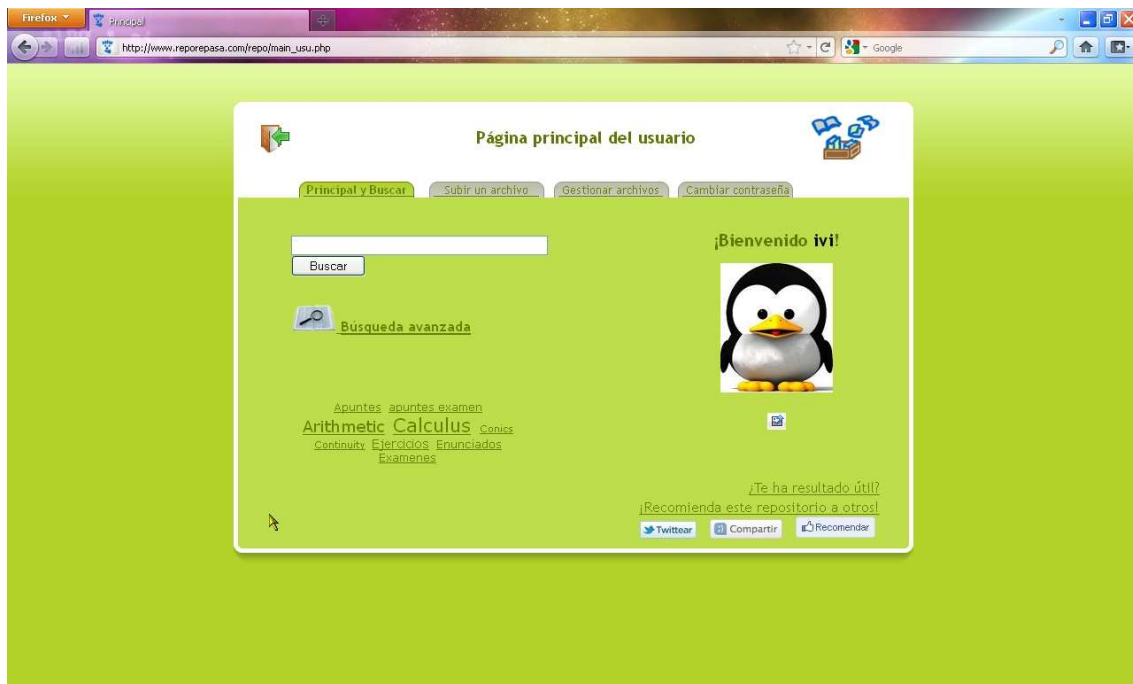


\*Si el usuario olvida su contraseña puede hacer clic en el link '¿Olvido el password?' e introducir su nombre de usuario y el email de contacto con el que se registró en el repositorio y se le enviará un correo recordándole su contraseña:





Una vez dentro de la aplicación, tenemos disponibles las siguientes funciones, las cuales se explicarán detalladamente:



## Buscar archivos

Disponemos de tres formas de buscar un archivo:



### - Búsqueda por similitud (tipo Google) (1):

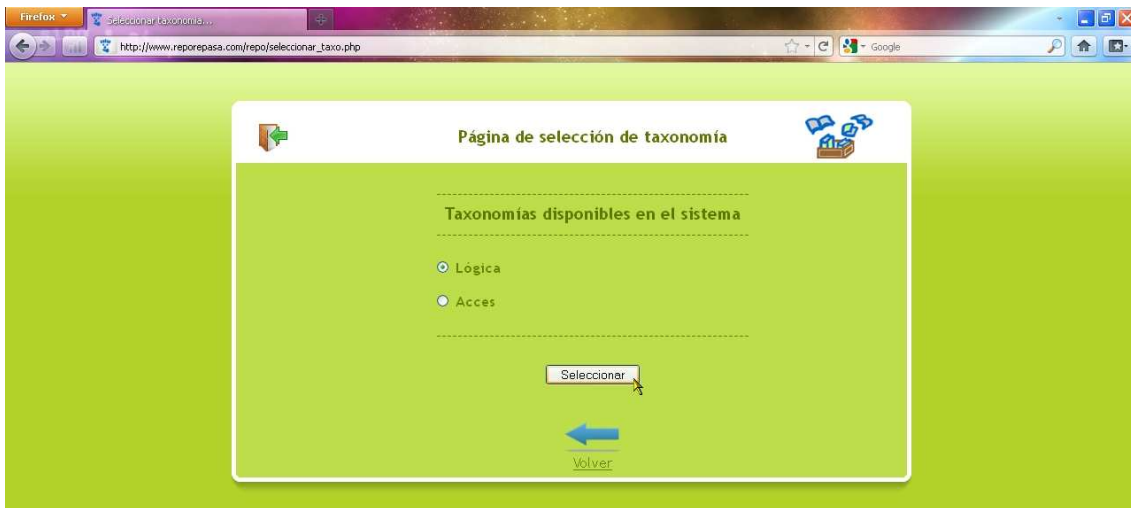
Basta con escribir una o varias palabras clave (los archivos tienen palabras clave, las cuales nos ayudarán a encontrarlos) y pulsar el botón “Buscar”:



### - Búsqueda avanzada (por etiquetas de la taxonomía) (2):

Ésta es la búsqueda por etiquetas de la taxonomía, por ello, tendremos primeramente que seleccionar la taxonomía con la que van a estar etiquetados los archivos que queremos encontrar. Seleccionamos la deseada y pulsamos el botón “Seleccionar”:





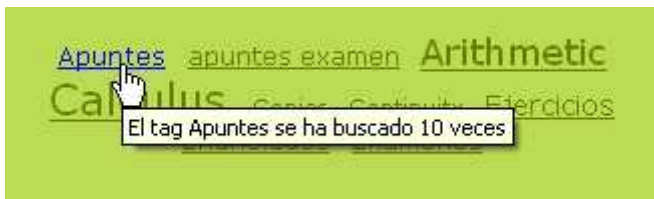
Ahora ya estamos dentro de la taxonomía seleccionada anteriormente y podemos ir navegando por ella seleccionando las etiquetas que necesitemos incluir en nuestra búsqueda, de modo que, cuando las tengamos, podremos pulsar el botón “Buscar”:



### - Búsqueda por etiqueta de la nube de tags (3):

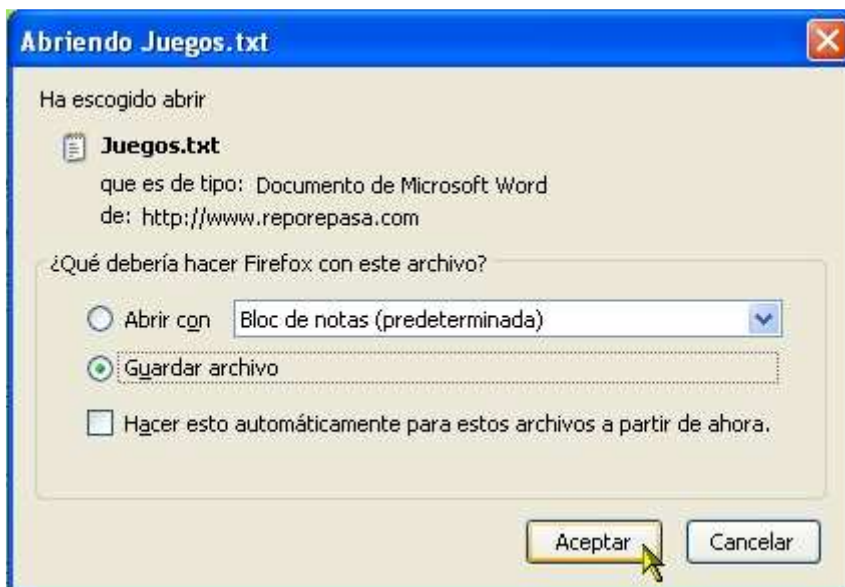
Cada vez que un usuario, registrado o no en el sistema, busca por una palabra clave o por una etiqueta, ésta se añade a la nube de tags si es de las 7 más buscadas, sino, quedará igualmente registrada pero no se mostrará en la nube.

De este modo, en la página principal de usuario tenemos acceso a las 7 etiquetas (tags) más buscadas por los usuarios, por tanto, con un sólo clic en un link de un tag nos lleva a los resultados de buscar esa palabra clave o etiqueta en todos los archivos:



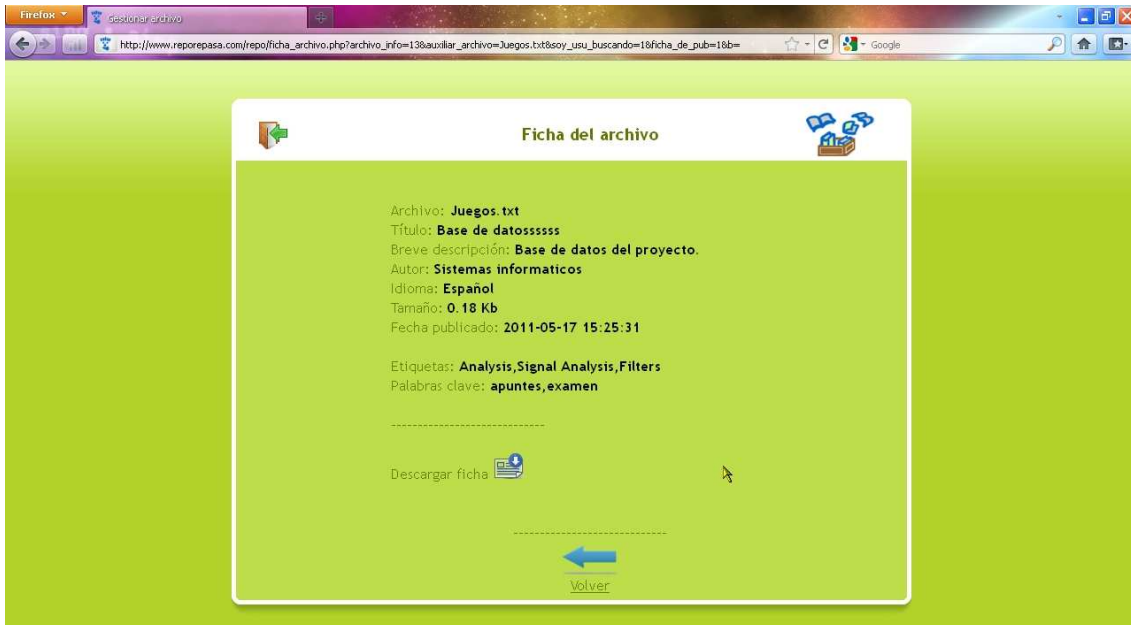
### Descargar archivo

Para la descarga de los archivos que hemos buscado basta con pulsar el icono de descarga:



## Ver/descargar ficha

Cuando buscamos un archivo, desde la página de resultados, podemos ver la ficha de un archivo pulsando el icono de ver ficha, que nos llevará a la página con la información de dicha ficha:



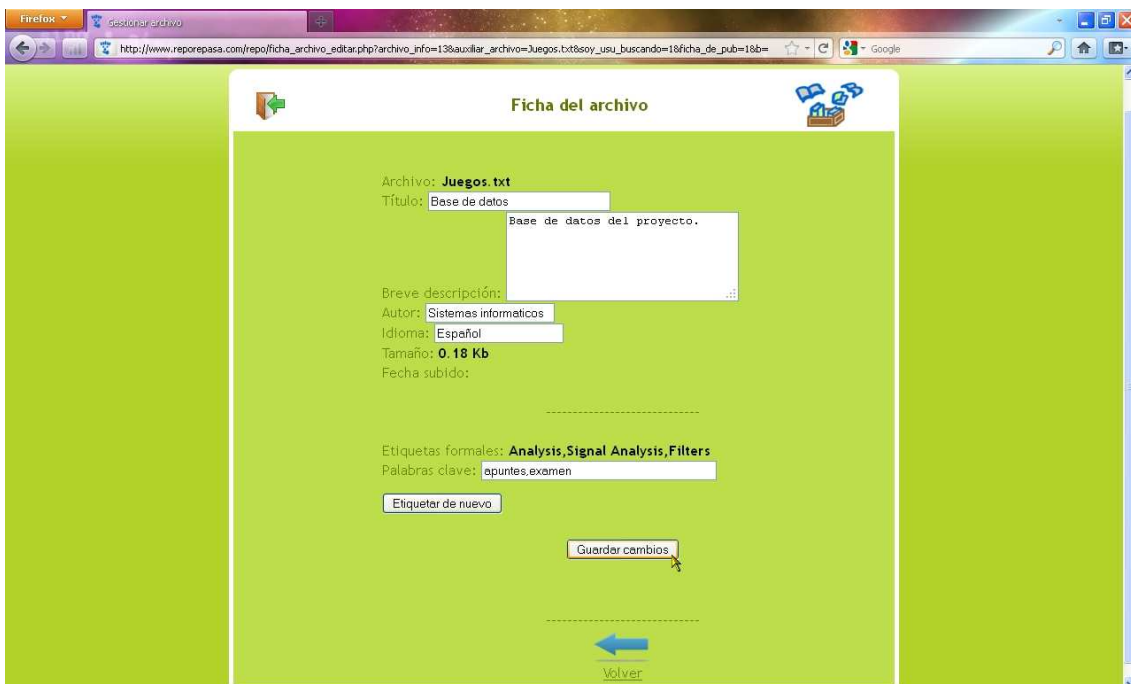
Desde aquí podremos descargar la ficha en un archivo de texto pulsando el icono de descargar ficha:



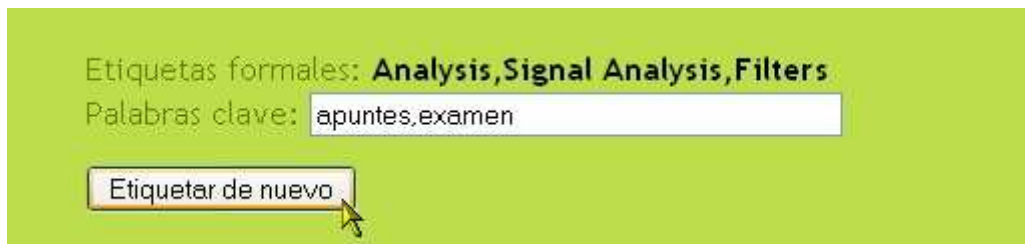


### Editar ficha

Pulsamos el icono de editar ficha y nos llevará a la información de la misma, la cual se podrá editar (sólo algunos campos). Una vez editada la información pulsaremos el botón “Guardar cambios”:

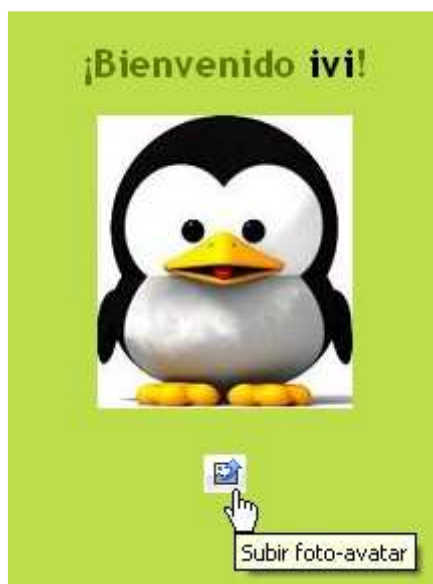


También tendremos la opción de etiqueta de nuevo el archivo, para ello, pulsaremos el botón “Etiquetar de nuevo” (ver Gestionar archivos - Etiquetar):



### Poner o cambiar el avatar

En la página principal del usuario se puede poner un avatar pulsando el siguiente botón:

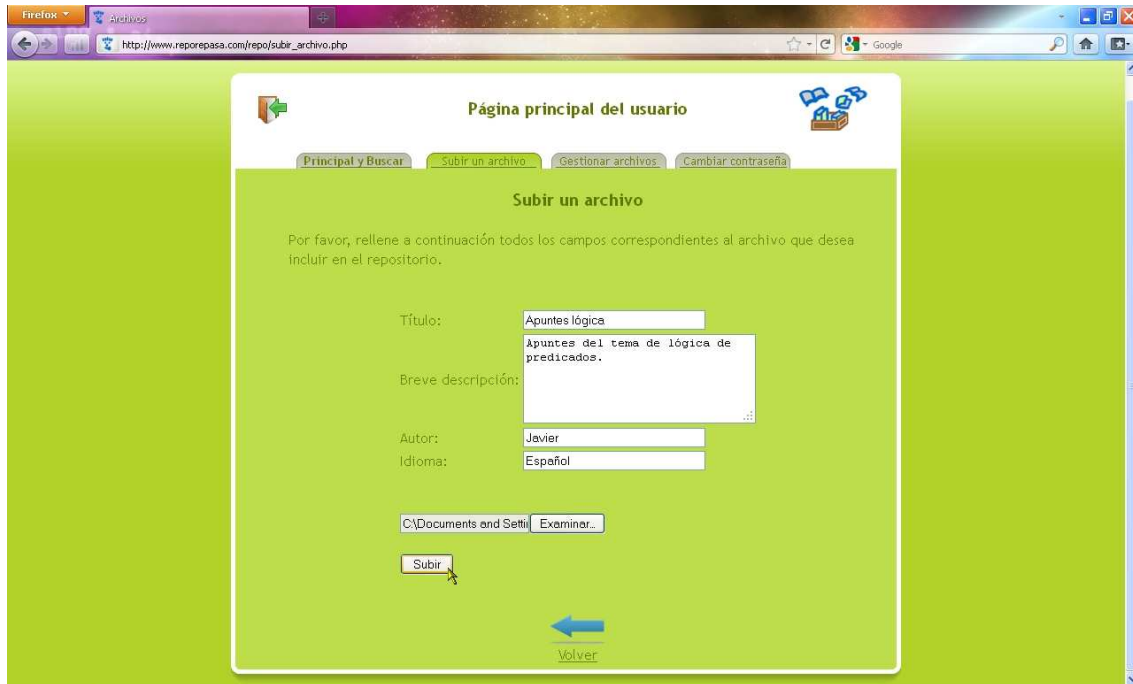


La imagen que podemos poner no debe superar los 30Kb de tamaño y debe ser un archivo con extensión .jpg. Una vez seleccionado pulsaremos “Subir”:



## Subir un archivo

Para subir un archivo al repositorio (será visible por el usuario que lo sube hasta que sea publicado), en la pestaña de subir un archivo, tenemos que rellenar todos los campos solicitados y seleccionar el archivo deseado y posteriormente pulsar el botón “Subir”:



The screenshot shows a web browser window with the URL [http://www.reporepasa.com/repo/subir\\_archivo.php](http://www.reporepasa.com/repo/subir_archivo.php). The page is titled "Página principal del usuario" and features a navigation menu with options: "Principal y Buscar", "Subir un archivo", "Gestionar archivos", and "Cambiar contraseña". The "Subir un archivo" option is selected.

The main content area is titled "Subir un archivo" and contains the following text: "Por favor, rellene a continuación todos los campos correspondientes al archivo que desea incluir en el repositorio."

The form fields are as follows:

- Título:
- Breve descripción:
- Autor:
- Idioma:

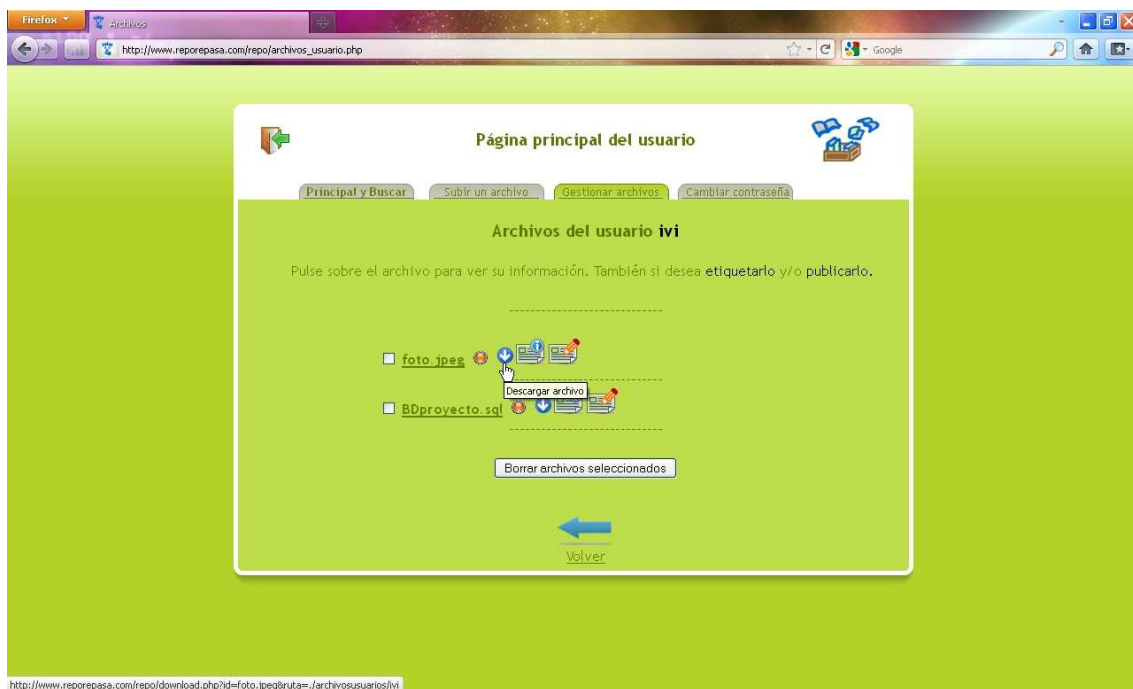
Below the form fields, there is a file selection button: . Below that is a "Subir" button, which is being clicked by a mouse cursor. At the bottom of the form area, there is a blue arrow pointing left with the text "Volver" below it.

## Gestionar archivos

Desde la gestión de archivos podremos realizar las siguientes operaciones:

### Descargar archivo

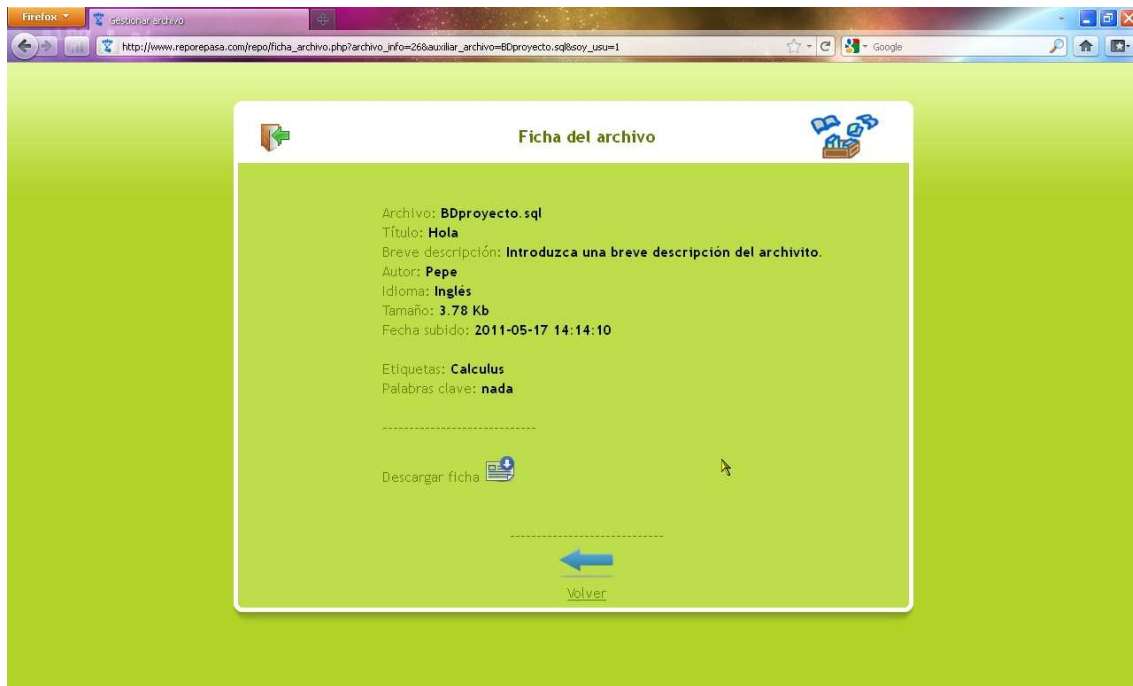
Para la descarga de los archivos que hemos buscado basta con pulsar el icono de descarga:



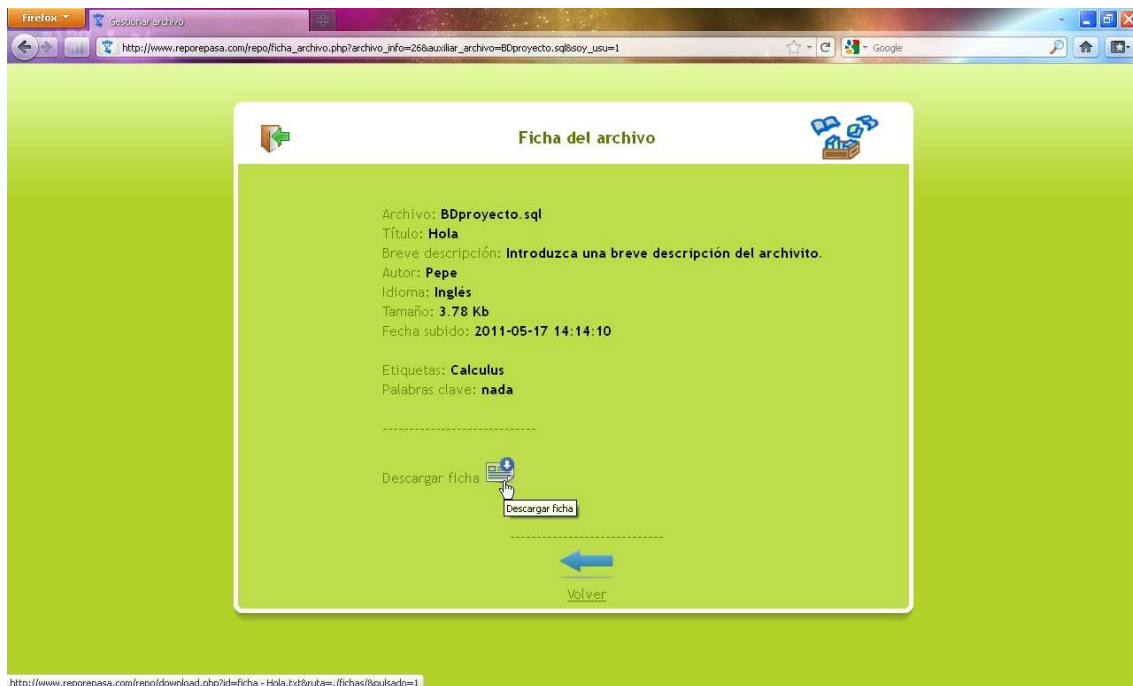
### Ver/descargar ficha

Cuando buscamos un archivo, desde la página de resultados, podemos ver la ficha de un archivo pulsando el icono de ver ficha, que nos llevará a la página con la información de dicha ficha:



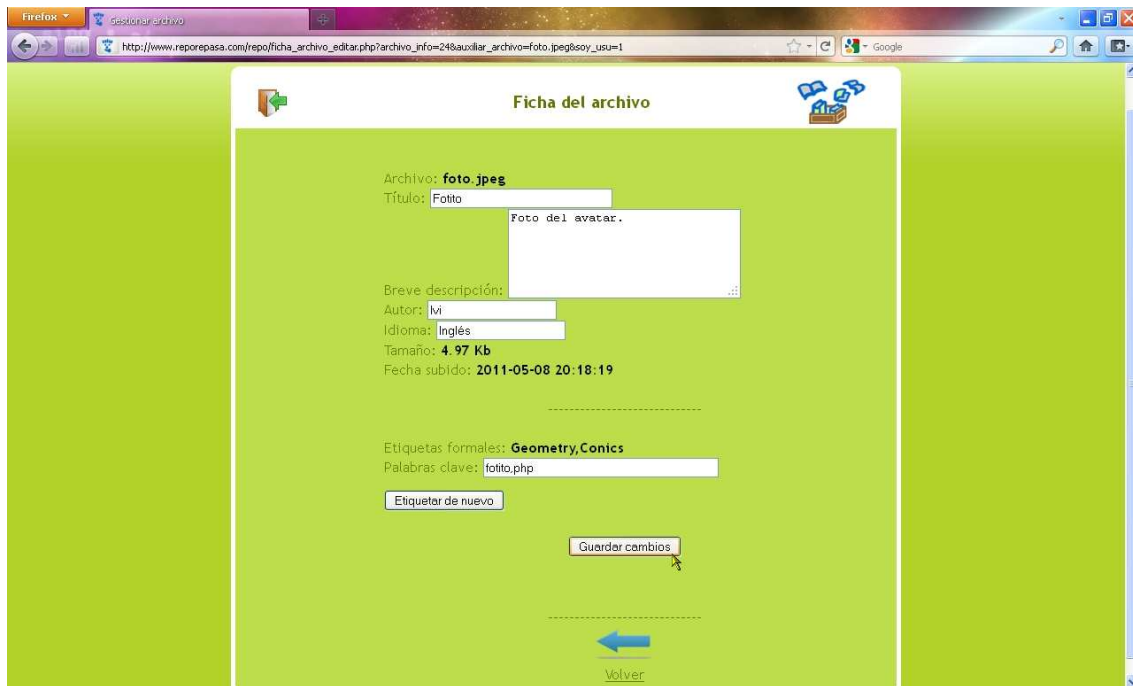


Desde aquí podremos descargar la ficha en un archivo de texto pulsando el icono de descargar ficha:

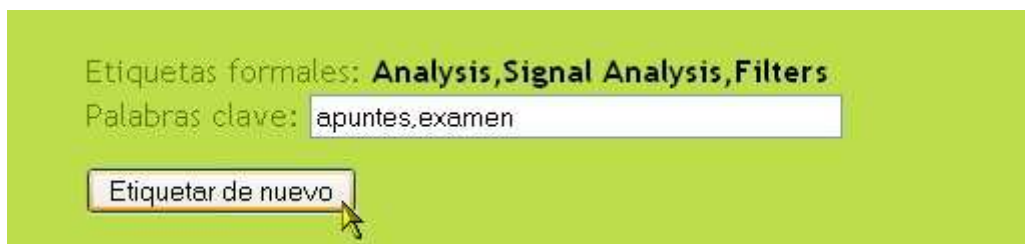


## Editar ficha

Pulsamos el icono de editar ficha y nos llevará a la información de la misma, la cual se podrá editar (sólo algunos campos). Una vez editada la información pulsaremos el botón “Guardar cambios”:



También tendremos la opción de etiqueta de nuevo el archivo, para ello, pulsaremos el botón “Etiquetar de nuevo” (ver Gestionar archivos - Etiquetar):

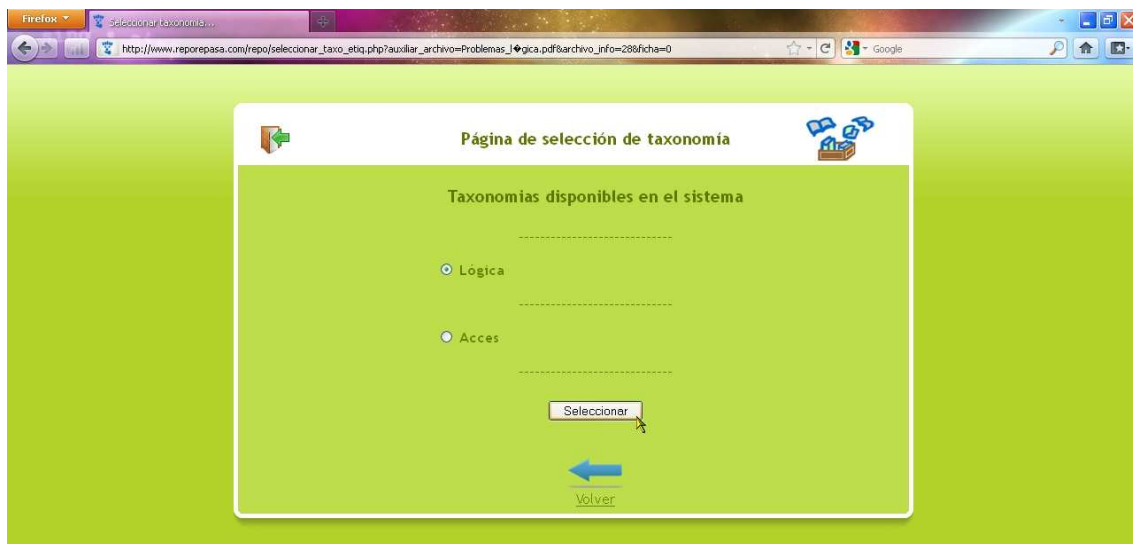
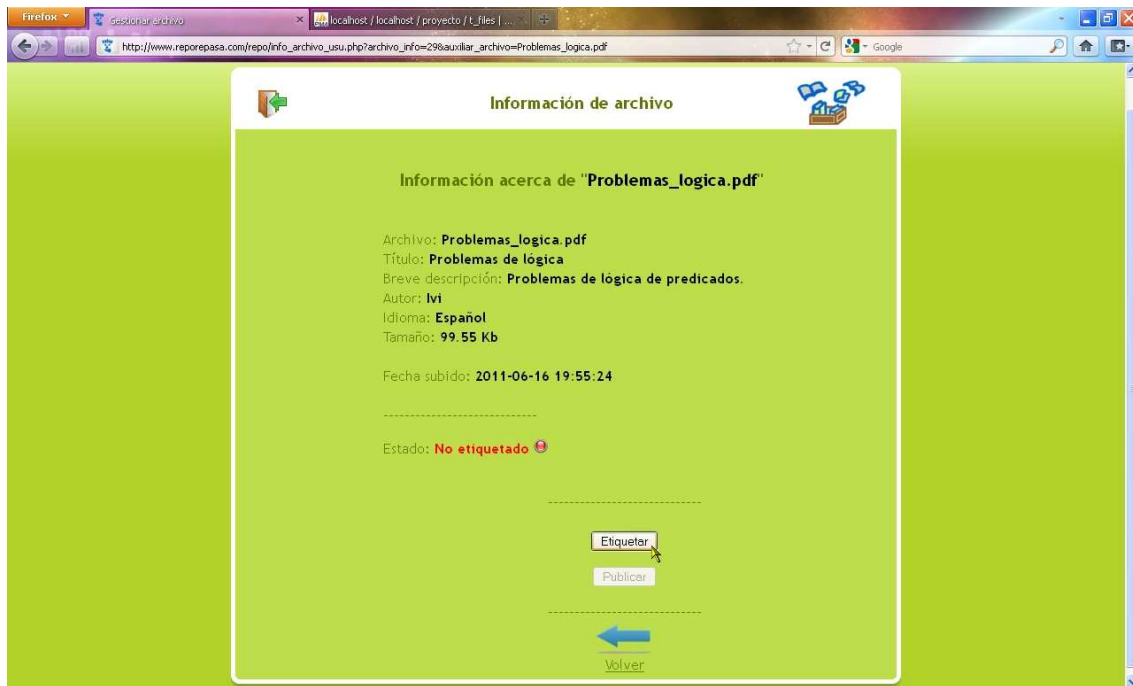


### Etiquetar archivo

Dentro de la información del archivo, para ello hemos pulsado el link con el nombre del mismo:



Para etiquetar un archivo, pulsamos el botón “Etiquetar” que nos llevará a la página de selección de taxonomía. Seleccionamos la taxonomía con la que vamos a etiquetar y pulsamos el botón “Seleccionar taxonomía”:



Ahora podremos ir navegando por las diferentes categorías de la taxonomía para ir construyendo la etiqueta formal. Cuando estemos dentro de la categoría con la que queremos marcar al archivo, pulsaremos el botón "Etiquetar":



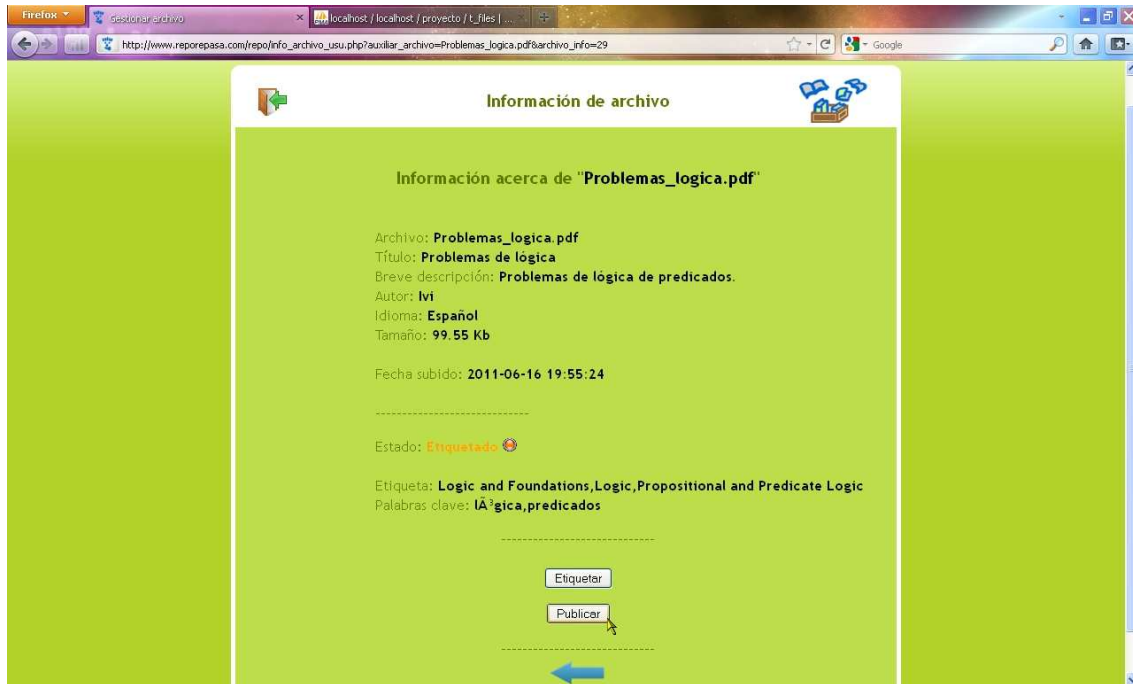
Éste, nos llevará al siguiente paso del etiquetado, las palabras clave. Para realizar este paso, escribiremos las palabras clave que queramos etiquetar al archivo, separadas por comas, y pulsaremos el botón “Etiquetar”:



Ahora ya tenemos el archivo completamente etiquetado, por lo que ya podemos publicarlo.

## Publicar archivo

Una vez que tenemos etiquetado el archivo, podremos pulsar el botón "Publicar" y de este modo todos los usuarios podrán acceder al archivo, pero el usuario que lo subió ya no tendrá acceso a él desde su página de gestión de archivos y no podrá borrarlo:



## Cambiar Contraseña

Es posible modificar la contraseña del usuario, sólo basta con introducir una nueva (por duplicado) y pulsar Aceptar:



## Salir de la aplicación

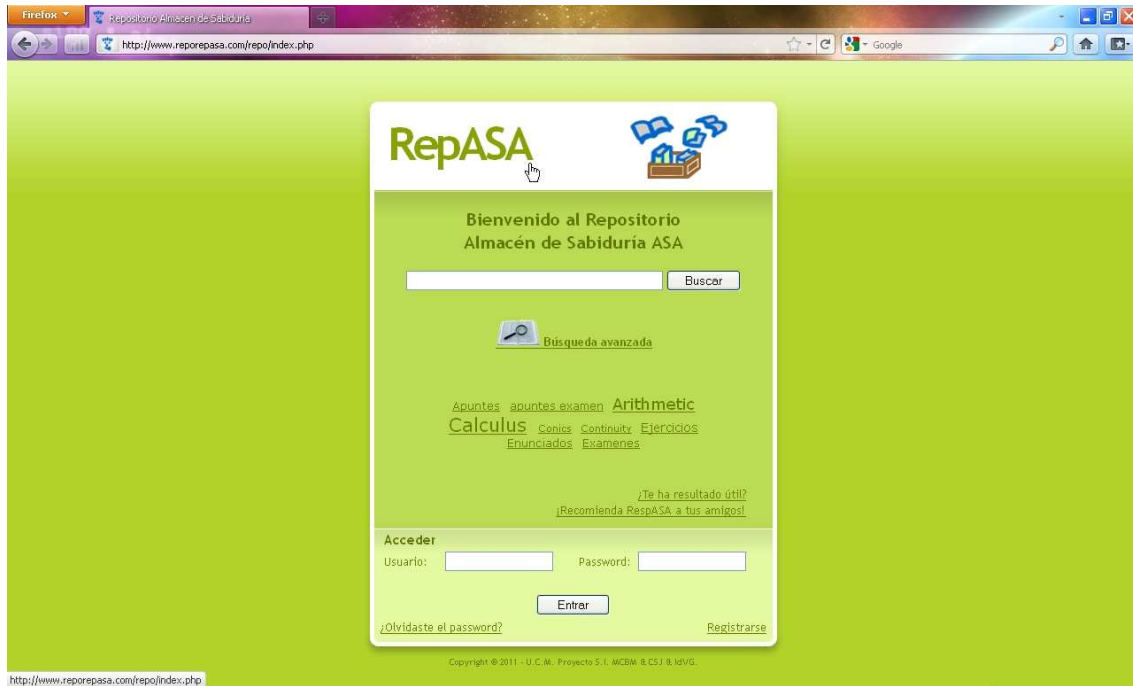
Para salir de la aplicación basta con pulsar el icono de 'Salir' (esquina superior izquierda):



## 6.5 Manual de explotación del usuario no registrado

Un usuario que no está registrado en el sistema podrá realizar búsquedas en el repositorio, descargarse los archivos que busca y ver/descargar las fichas de los mismos.

Desde la página principal de RepASA:



Podremos realizar diversas funciones que explicaremos a continuación:

## Registrarse

Para registrarse en el repositorio RepASA, basta con hacer clic en el link 'Registrarse' (esquina inferior derecha) y rellenar todos los campos que se solicitan:



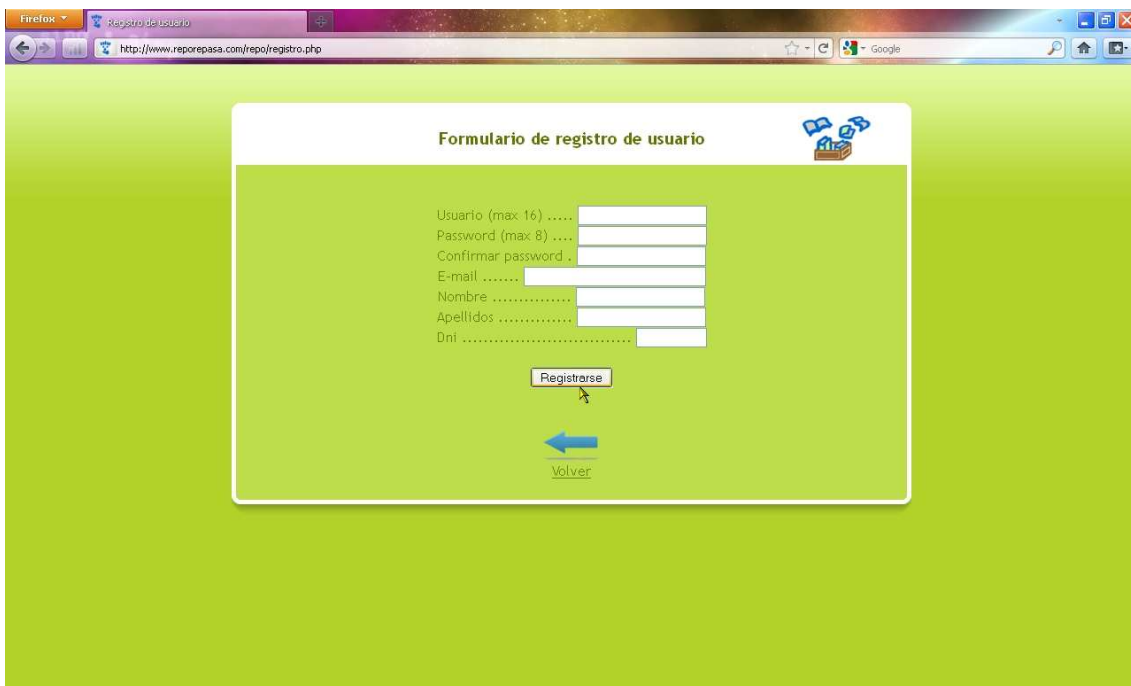
Acceder

Usuario:  Password:

Entrar

[¿Olvidaste el password?](#) [Registrarse](#)

Copyright © 2011 - U.C.M. Proyecto S.I. MCBM & CSJ & IdVG.



Firefox - Registro de usuario

http://www.reporepasa.com/epo/registro.php

Formulario de registro de usuario

Usuario (max 16) .....

Password (max 8) .....

Confirmar password .....

E-mail .....

Nombre .....

Apellidos .....

Dni .....

Registrarse

Volver

Una vez completado el registro, el usuario podrá acceder introduciendo su usuario y contraseña en la página principal:



Acceder

Usuario:  Password:

Entrar

[¿Olvidaste el password?](#) [Registrarse](#)

Copyright © 2011 - U.C.M. Proyecto S.I. MCBM & CSJ & IdVG.

## Buscar un archivo

Disponemos de tres formas de buscar un archivo:



### - Búsqueda por similitud (tipo Google) (1):

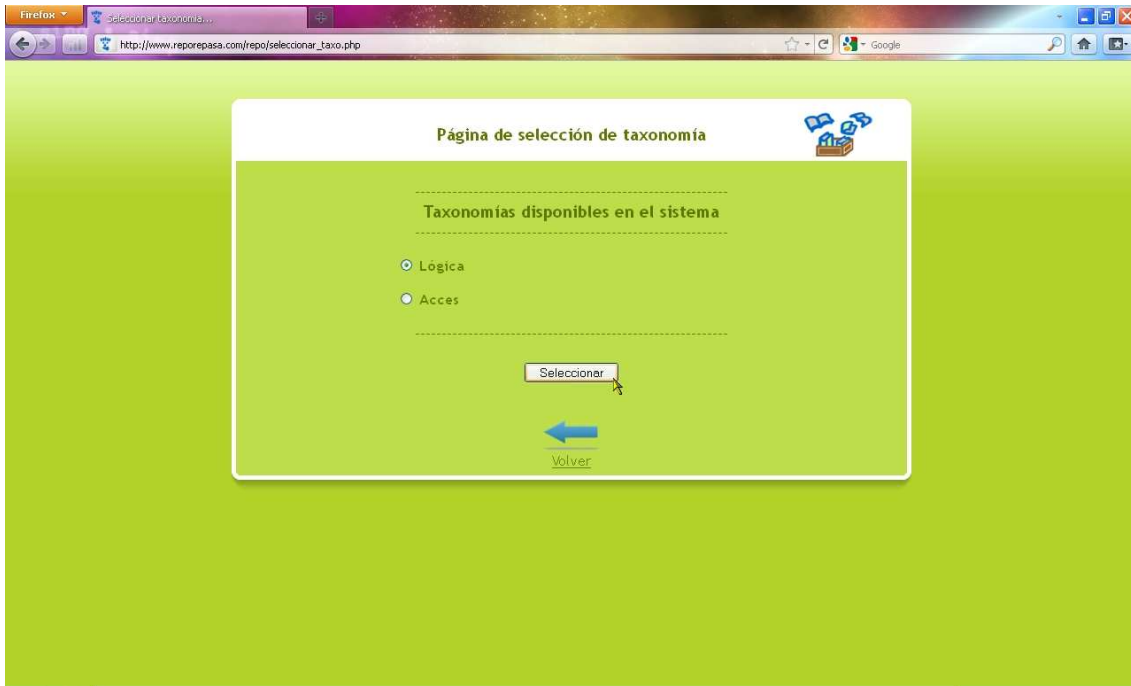
Basta con escribir una o varias palabras clave (los archivos tienen palabras clave, las cuales nos ayudarán a encontrarlos) y pulsar el botón “Buscar”:



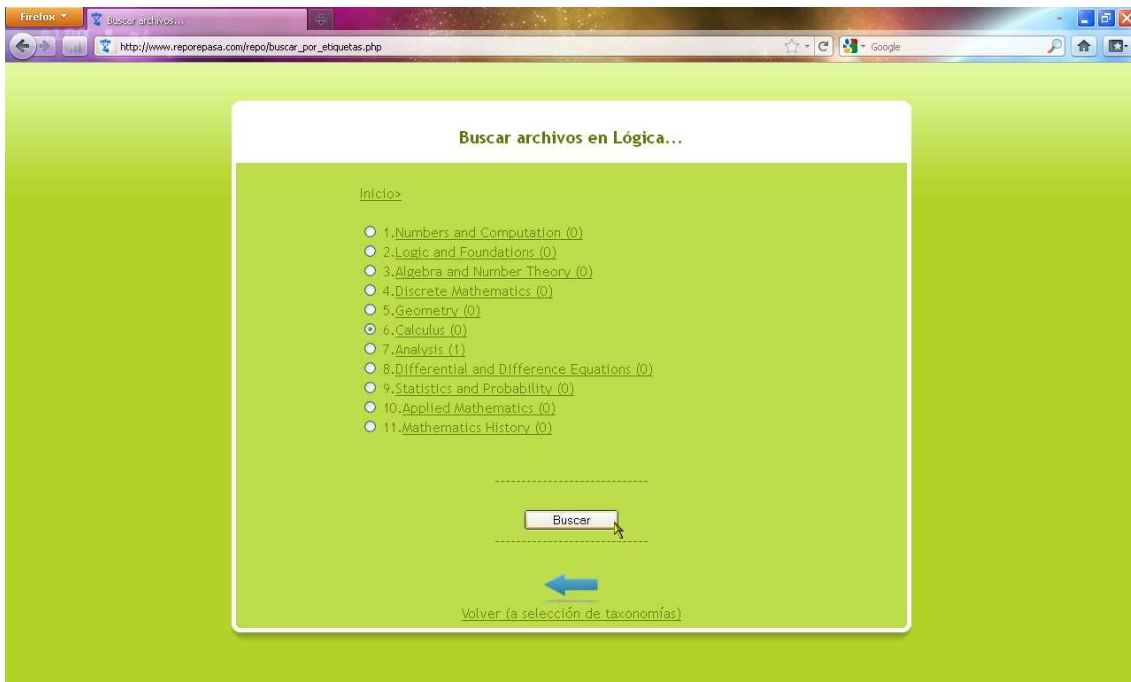
### - Búsqueda avanzada (por etiquetas de la taxonomía) (2):

Ésta es la búsqueda por etiquetas de la taxonomía, por ello, tendremos primeramente que seleccionar la taxonomía con la que van a estar etiquetados los archivos que queremos encontrar. Seleccionamos la deseada y pulsamos el botón “Seleccionar”:





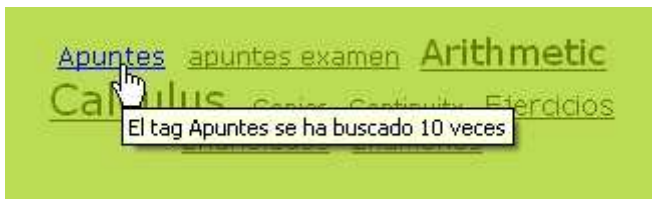
Ahora ya estamos dentro de la taxonomía seleccionada anteriormente y podremos ir navegando por ella seleccionando las etiquetas que necesitemos incluir en nuestra búsqueda, de modo que, cuando las tengamos, podremos pulsar el botón “Buscar”:



### - Búsqueda por etiqueta de la nube de tags (3):

Cada vez que un usuario, registrado o no en el sistema, busca por una palabra clave o por una etiqueta, ésta se añade a la nube de tags si es de las 7 más buscadas, sino, quedará igualmente registrada pero no se mostrará en la nube.

De este modo, en la página principal tenemos acceso a las 7 etiquetas (tags) más buscadas por los usuarios, por tanto, con un sólo clic en un link de un tag nos lleva a los resultados de buscar esa palabra clave o etiqueta en todos los archivos:



### Descargar un archivo

Para descargar un archivo, que hemos buscado previamente, tenemos dos formas de hacerlo:

- Desde la página de resultados de la búsqueda, pulsamos el icono de descargar archivo:



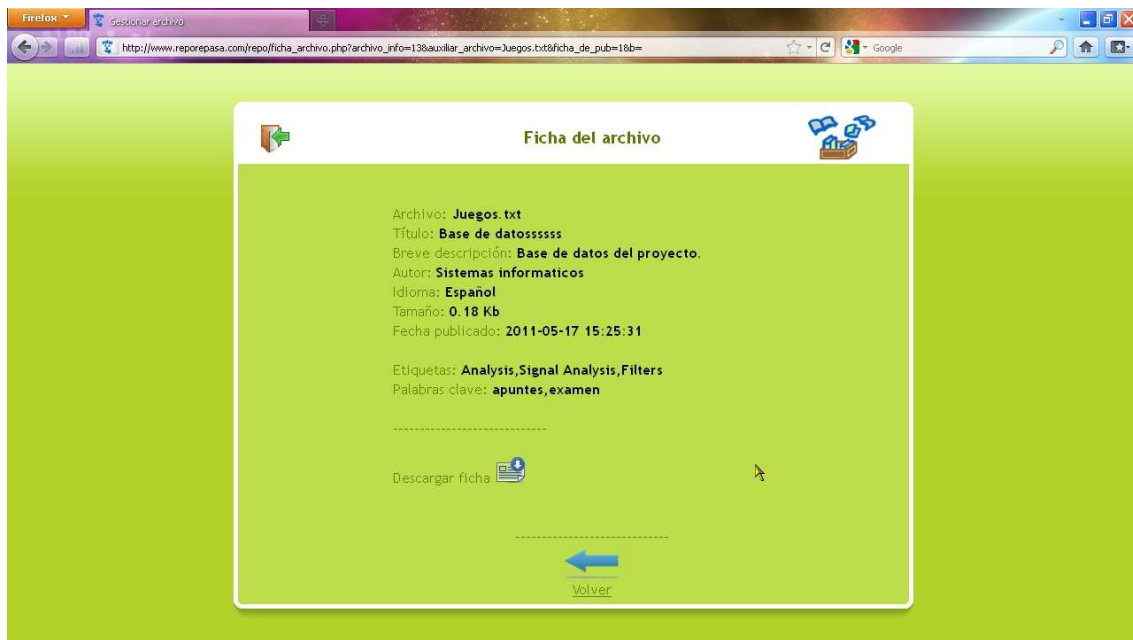


- Desde la página de resultados, pulsamos el link con el nombre del fichero y vamos a la página de información del archivo y desde aquí podemos descargarlo presionando el botón “Descargar”:

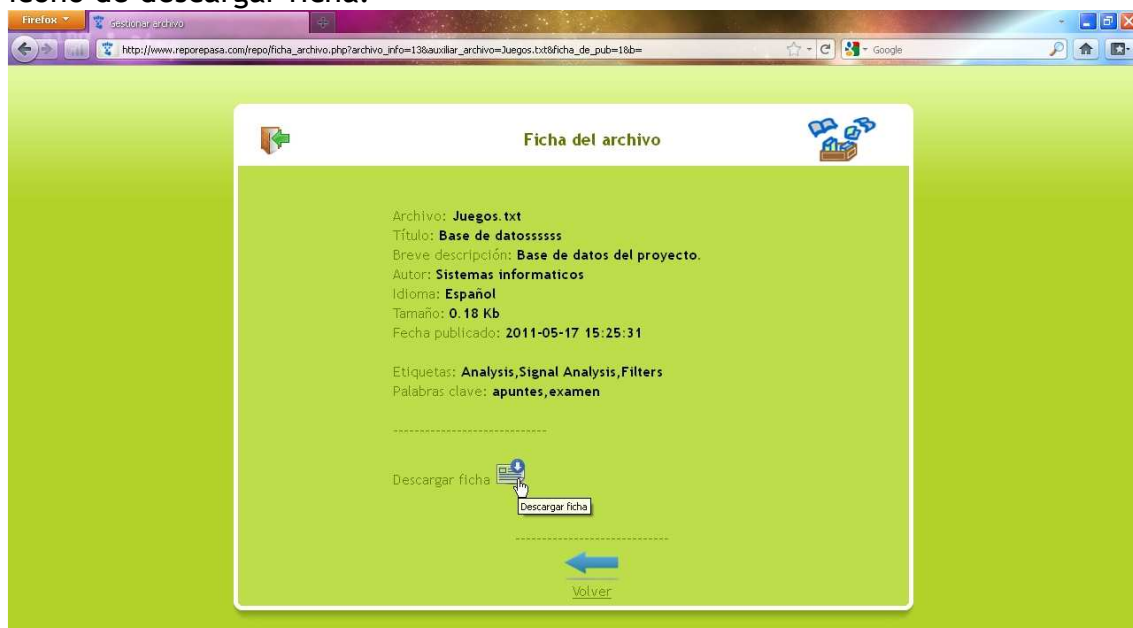


## Ver/descargar ficha de archivo

Cuando buscamos un archivo, desde la página de resultados, podemos ver la ficha de un archivo pulsando el icono de ver ficha, que nos llevará a la página con la información de dicha ficha:



Desde aquí podremos descargar la ficha en un archivo de texto pulsando el icono de descargar ficha:



## Capítulo 7: Dificultades, conclusiones y trabajo futuro

La principal dificultad encontrada desde el comienzo fue nuestro total desconocimiento del lenguaje de programación PHP y del trabajo con servidores.

Estos problemas que tuvimos durante los inicios, sumados al hallazgo de XAMPP, nos hicieron decantarnos por usar este amigable programa para la instalación de Apache, MySQL y PHPMyAdmin.

Como venimos repitiendo, RepASA es un repositorio robusto y sencillo. Características que no son en absoluto negativas (*"Keep it simple!"*).

\* El principio KISS (*"Keep It Short and Simple"*) recomienda el desarrollo de sistemas empleando partes sencillas, comprensibles y con errores de fácil detección y corrección, rechazando lo enrevesado e innecesario.

Tiene las funcionalidades básicas de un buen repositorio pero desde un comienzo tuvimos en mente que serviría para añadirle futuras mejoras.

Algunas de estas mejoras, ampliaciones y trabajo futuro serían:

- Actualmente puedes descargar un archivo o su ficha de forma independiente. Una futura gran mejora sería que ambas cosas se juntaran en un único archivo comprimido. Este paquete sería lo que se llama "objeto de aprendizaje" y la mejora sería guardarlo y etiquetarlo de forma normalizada (IMS Card Reading, IMSCP (IMS Content Packaging), SCORM (Sharable Content Object Reference Model)).
- Actualmente las listas de archivos y las listas de usuarios en la página web pueden ser tan largas como permita la base de datos que hay detrás. Gracias a la forma en que confeccionamos el característico recuadro blanco de RepASA, esto no es problema; porque, cuando la página vaya creciendo hacia abajo, el borde de los laterales lo hará también y el aspecto seguirá impecable. Pero, cuando haya muchos usuarios es muy tedioso tener que mover un 'scroll' larguísimo, así que la mejora sería paginar las listas; es decir, mostrar sólo unos cuantos resultados y añadir unos botones "anterior", "siguiente", "ir al principio" e "ir al final", que permitieran mostrar los 'n' siguientes elementos.
- En línea con lo anterior, dado que las listas pueden ser muy largas, una posible mejora pensando en la accesibilidad y los discapacitados sería incluir una forma de saltar una lista completa cuando naveguen ayudados de tecnologías que se guíen por el foco de escritura (es decir, como lo hace la tecla 'tabulador' cuando navegas con el teclado).

- Una asunto menos importante pero que, aún así merece mencionarse es que, en el código, debido a ese inicial desconocimiento que comentábamos, hay muchas líneas que son completamente de HTML que están entre los tags que indican dónde comienza el código PHP; por lo que el servidor está “interpretando” esas líneas y comprobando si tiene que ejecutar algún fragmento PHP, perdiendo, así, eficiencia. La mejora, para ganar eficiencia y también legibilidad y facilidad de comprensión del código para alguien que lo lee por primera vez, sería buscar las líneas “echo (.....)” de PHP que lo único que hagan sea imprimir código HTML, y cambiarlas por ese código que llevan dentro.
- Otra posible ampliación que podría plantearse es que, ya que gran parte del texto (nombre de los archivos, etiquetas, nombre de los usuarios...) se saca de la base de datos, traducir el resto del texto del repositorio (los títulos de las páginas y las frases que van fijadas en el código) a otros idiomas. De forma que con no demasiado trabajo se convertiría en un repositorio más plural y general todavía de lo que ya es.

## Bibliografía

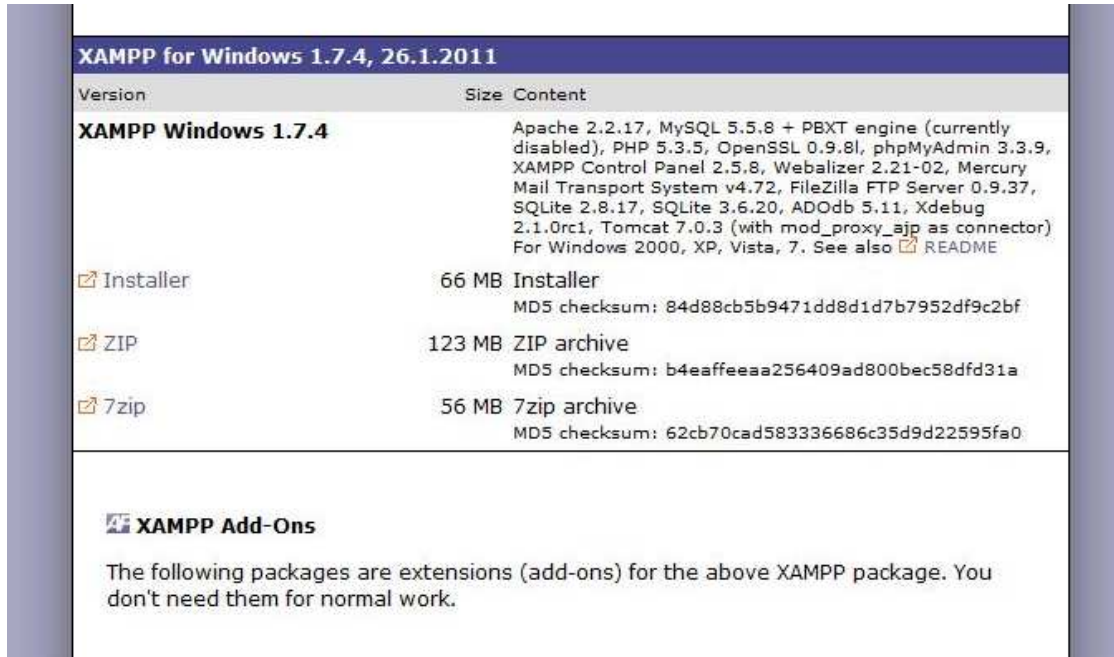
- [1] **Agrega:** <http://www.proyectoagrega.es/default/Documentacion>
- [2] **LOM:** [http://es.wikipedia.org/wiki/Archivo:LOM\\_base\\_schema.png](http://es.wikipedia.org/wiki/Archivo:LOM_base_schema.png)
- [3] **Repositorios:** [www.dlib.org/dlib/january10/reilly/01reilly.html](http://www.dlib.org/dlib/january10/reilly/01reilly.html)
- [4] **Idea de la plantilla web:** <http://www.freecsstemplates.org>
- [5] **Antonio Sarasa Cabezuelo:** Desarrollo de un sistema de clasificación de recursos electrónicos en el ámbito de la lógica matemática.
- [6] **Lorenzo García Arieto:** Objetos de aprendizaje, características y repositorios.
- [7] **XML:** <http://www.w3.org/xml>
- [8] **Dublin Core:** <http://dublincore.org/>
- [9] **Gloria Elena Vásquez:** Escritura científica y repositorios de acceso abierto
- [10] **MERLOT:** <http://www.merlot.org/>
- [11] **Fedora:** <http://www.fedoraproject.org/>
- [12] **Careo:** <http://www.careo.org/>



# Anexo 1: Guía de instalación de XAMPP y Repositorio

1.- Entramos en la página de XAMPP:  
<http://www.apachefriends.org/es/xampp.html>

2.- En nuestro caso, buscamos la versión para Windows...

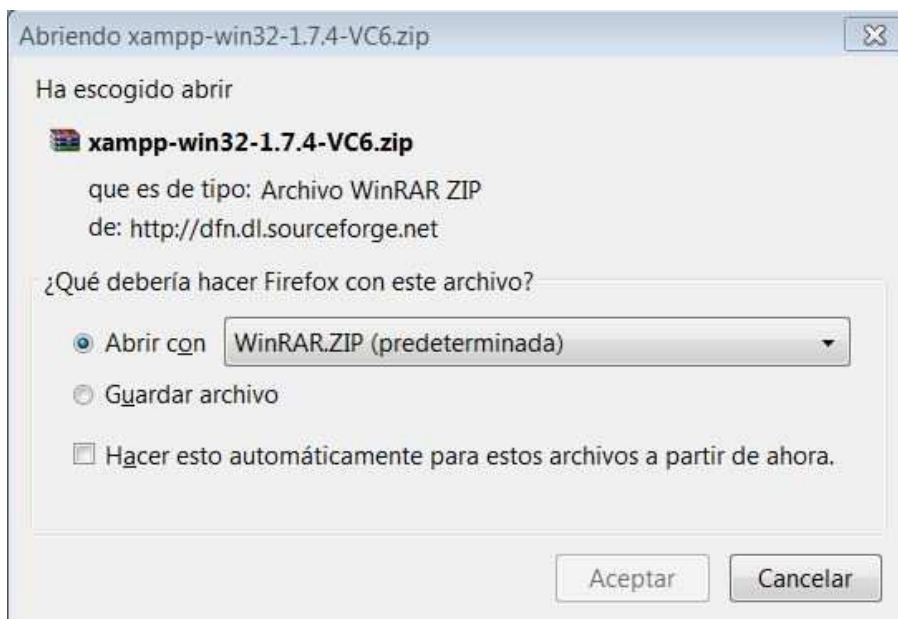


Version	Size	Content
<b>XAMPP Windows 1.7.4</b>		Apache 2.2.17, MySQL 5.5.8 + PBXT engine (currently disabled), PHP 5.3.5, OpenSSL 0.9.8l, phpMyAdmin 3.3.9, XAMPP Control Panel 2.5.8, Webalizer 2.21-02, Mercury Mail Transport System v4.72, FileZilla FTP Server 0.9.37, SQLite 2.8.17, SQLite 3.6.20, ADOdb 5.11, Xdebug 2.1.0rc1, Tomcat 7.0.3 (with mod_proxy_ajp as connector) For Windows 2000, XP, Vista, 7. See also <a href="#">README</a>
<a href="#">Installer</a>	66 MB	Installer MD5 checksum: 84d88cb5b9471dd8d1d7b7952df9c2bf
<a href="#">ZIP</a>	123 MB	ZIP archive MD5 checksum: b4eaffeeaa256409ad800bec58dfd31a
<a href="#">7zip</a>	56 MB	7zip archive MD5 checksum: 62cb70cad583336686c35d9d22595fa0

**XAMPP Add-Ons**

The following packages are extensions (add-ons) for the above XAMPP package. You don't need them for normal work.

3.- Elegimos para descargar el archivo '.zip'...



Abriendo xampp-win32-1.7.4-VC6.zip

Ha escogido abrir

**xampp-win32-1.7.4-VC6.zip**  
que es de tipo: Archivo WinRAR ZIP  
de: http://dfn.dl.sourceforge.net

¿Qué debería hacer Firefox con este archivo?

Abrir con WinRAR.ZIP (predeterminada)

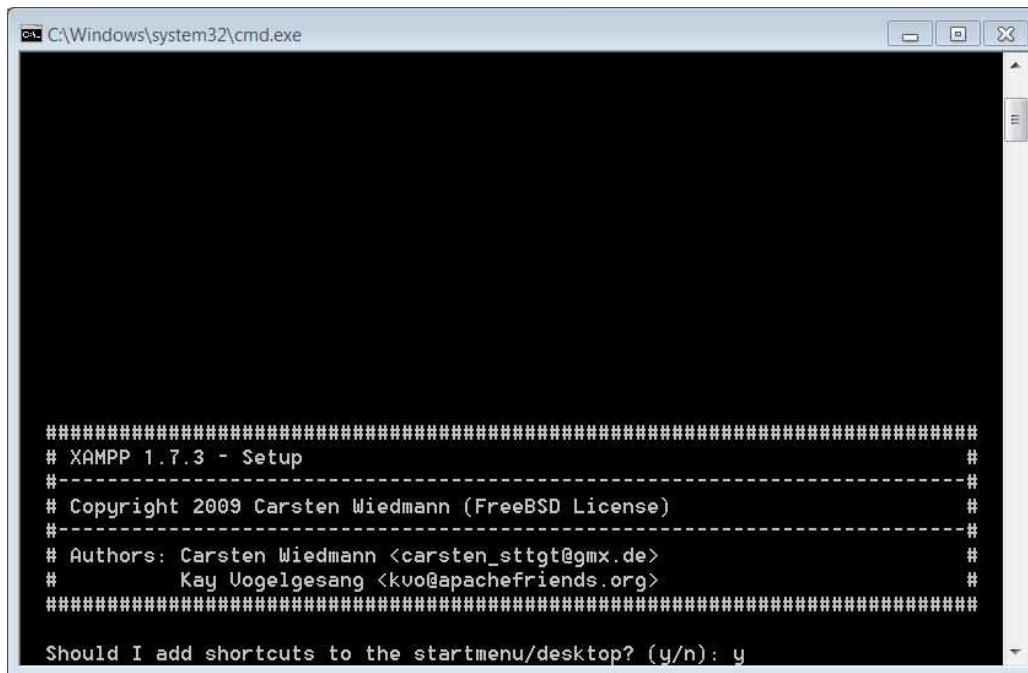
Guardar archivo

Hacer esto automáticamente para estos archivos a partir de ahora.

Aceptar Cancelar

4.- Extraer el contenido del '.zip' en "C:" (o en cualquier unidad mientras esté en la raíz). Lo que nos deja la carpeta de la forma C:\xampp.

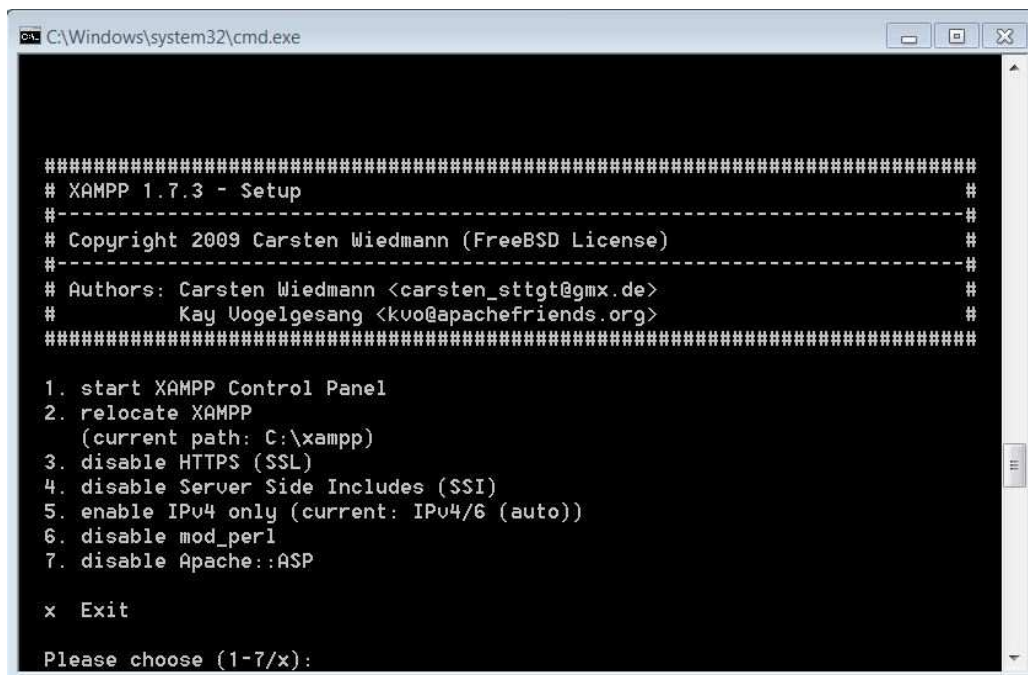
5.- Para comenzar la instalación, ejecutar "setup\_xampp.bat"  
(XAMPP no creará entradas ni nuevas variables en el registro de Windows)



```
C:\Windows\system32\cmd.exe

#####
# XAMPP 1.7.3 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)     #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>     #
#           Kay Vogelgesang <kvo@apachefriends.org>      #
#####
Should I add shortcuts to the startmenu/desktop? (y/n): y
```

6.- Nos irá preguntando una serie de acciones a realizar o no; podemos aceptar los valores que tiene por defecto dando a ENTER. Tras cinco preguntas, habrá terminado y, cuando nos pregunte en este último menú que qué deseamos hacer, le decimos con la 'x' que 'cerrar'.



```
C:\Windows\system32\cmd.exe

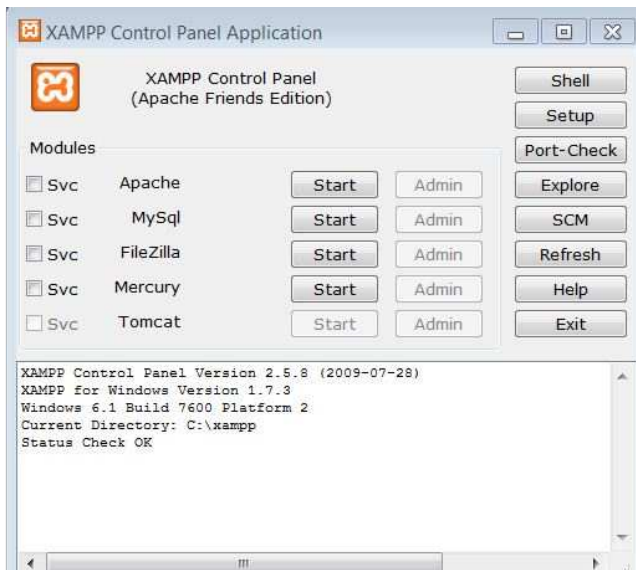
#####
# XAMPP 1.7.3 - Setup                                     #
#-----#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)     #
#-----#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>     #
#           Kay Vogelgesang <kvo@apachefriends.org>      #
#####

1. start XAMPP Control Panel
2. relocate XAMPP
   (current path: C:\xampp)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 (auto))
6. disable mod_perl
7. disable Apache::ASP

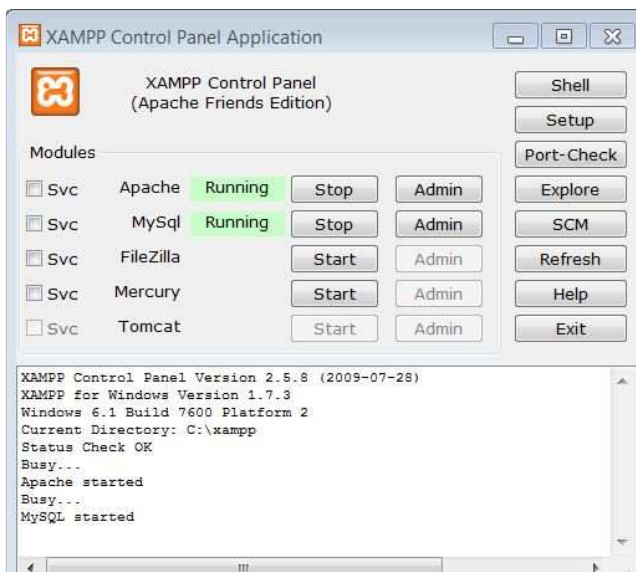
x Exit

Please choose (1-7/x):
```

7.- Cuando termina la instalación, para iniciarlo le damos doble clic a 'xampp-control.exe'. Lo cual abre esta ventana:

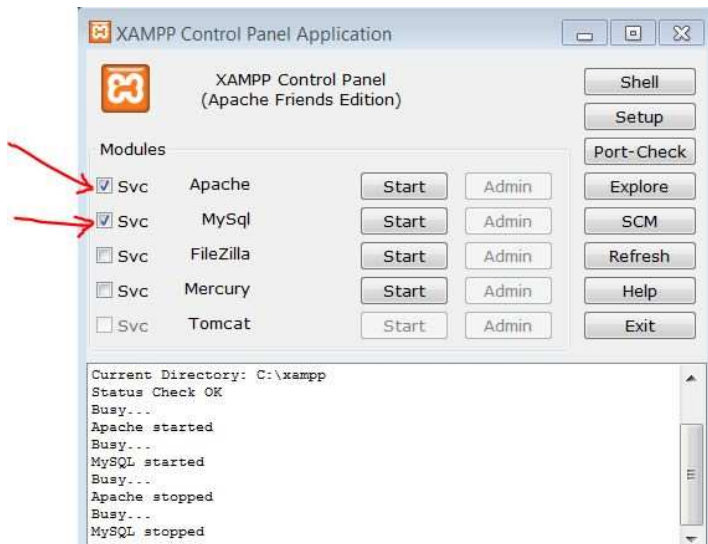


8 - Le damos a “Start” a Apache y a “Start” a MySQL.  
(Si sale algún aviso del firewall de Windows, le damos a “permitir acceso”).



(También puedes iniciarlos con \xampp\apache\_start.bat y \xampp\mysql\_start.bat)

**NOTA:** cada vez que apagues el ordenador, para, al encenderlo, volver a usar tu máquina como servidor, tendrás que iniciar de nuevo “Apache” y “MySQL”. Si no quieres tener que hacer esto todas las veces, puedes dejar que se ejecuten como “servicios”; esto lo consigues al marcar la ‘checkbox’ de cada uno de ellos (flechas rojas de la siguiente imagen), cuando estén detenidos.



9.- Abrimos el navegador y escribimos en la barra de direcciones <http://127.0.0.1/> o <http://localhost/> , lo cual nos llevará a esta página:



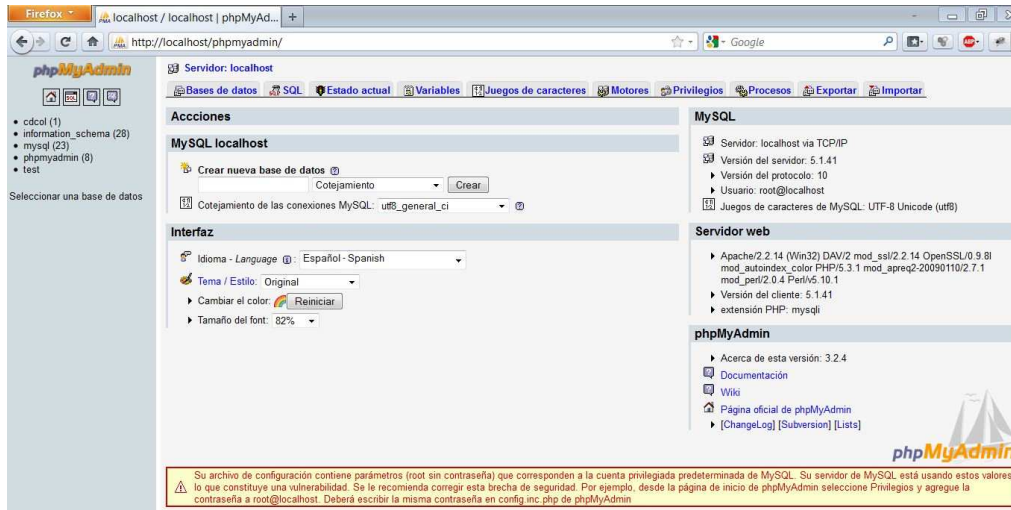
English / Deutsch / Français / Nederlands / Polski / Slovens / Italiano / Norsk / Español / 中文 / Português / Português (Brasil) / 日本語

10.- Elegimos idioma español y nos lleva a esta otra página:

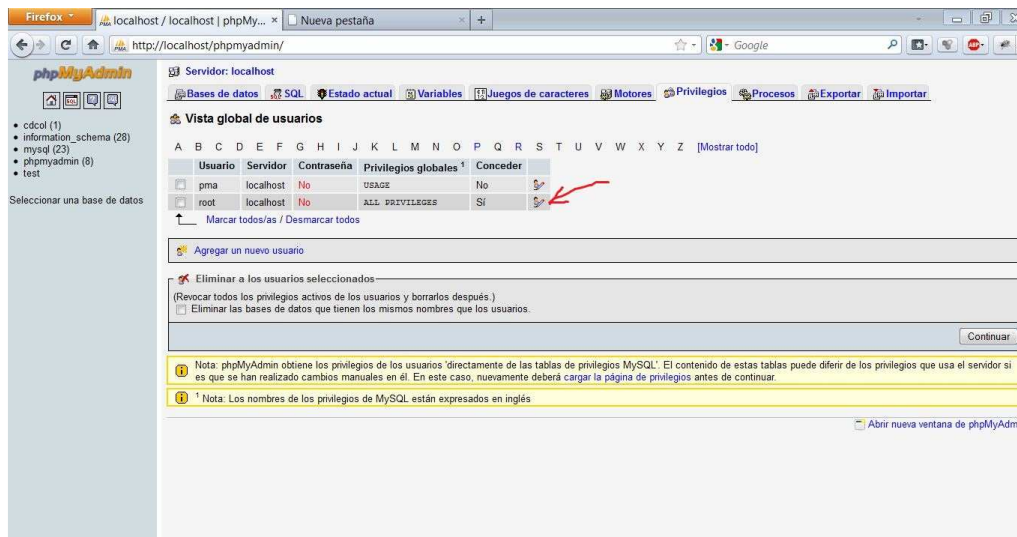


11 - En el menú de la izquierda buscamos “phpMyAdmin” y hacemos clic.

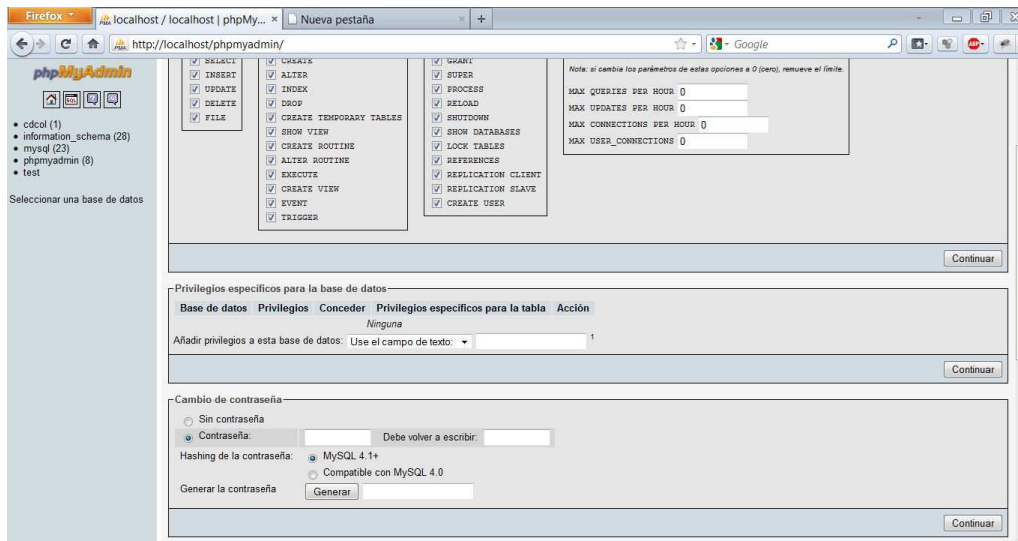
12 - Al entrar en phpMyAdmin, llegas a esto:



13.- Para ponerle al usuario root la contraseña de nuestro proyecto, vamos a la pestaña “privilegios” y, en root (marcado con la flecha roja), “editar”.

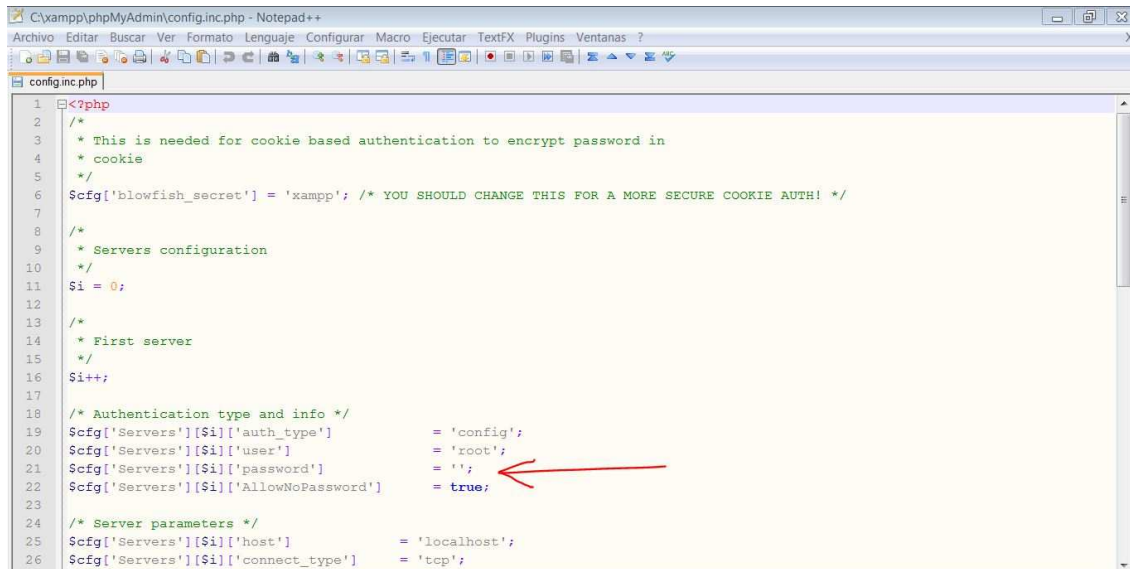


14.- Bajamos a ‘cambio de contraseña’, ponemos la nuestra (root@123) y damos a ‘continuar’.

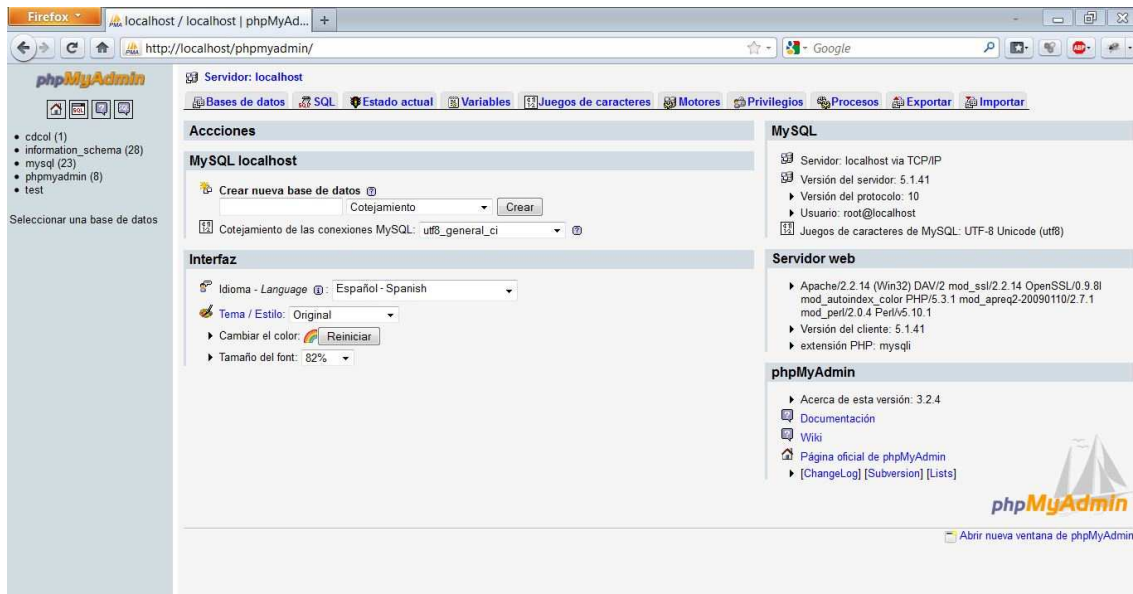


15.- Para que el password que acabamos de cambiar en la base de datos quede reflejado también en el archivo de configuración de nuestra instalación de PHP, nos vamos a “C:\xampp\phpMyAdmin” y abrimos con un editor el archivo “config.inc.php”

Allí, en donde pone “password” (marcado en la siguiente imagen con una flecha roja), escribimos el mismo que antes: root@123



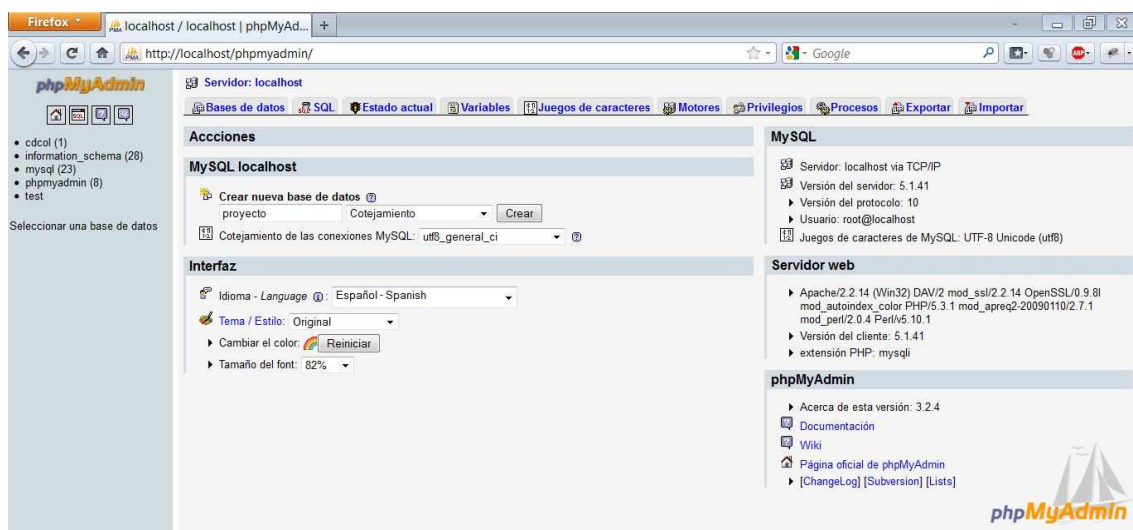
16.- Ahora podemos actualizar la página de phpMyAdmin sin que reporte ningún error y podemos seguir adelante.



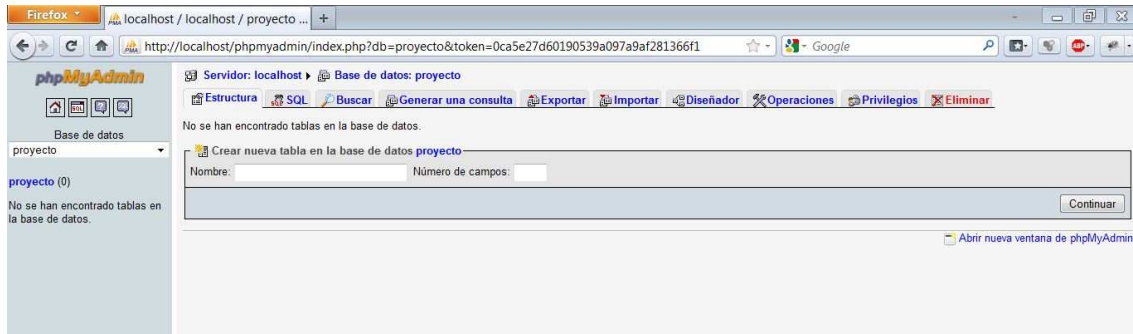
17.- Del archivo “.zip” que contiene el código de nuestro proyecto, extraemos la carpeta “repo” y la pegamos dentro de “C:/xampp/htdocs”.

NOTA: Con XAMPP, el directorio raíz (y archivo ‘index’) para los HTTP es → \xampp\htdocs.

18.- Otra vez en phpMyAdmin, creamos una nueva base de datos de nombre “proyecto”.



19.- Nos vamos a la vista de esa nueva base de datos (haciendo clic en su nombre en el menú de la izquierda) y, desde ahí, importamos (sexta pestaña) el archivo ‘.sql’ que tenemos en “C:/xampp/htdocs/repo” y le damos a “continuar”.



20.- Volvemos al navegador. Poniendo en la barra de direcciones: “http://localhost/repo”, ahí tenemos ya nuestro repositorio listo para usarse.



**NOTA:** Desinstalar XAMPP sería simplemente eliminar por completo el directorio “xampp”.

## Anexo 2: Ejemplo de taxonomía

Estructura de la taxonomía que se ha utilizado durante el desarrollo y las pruebas del proyecto, obtenida del trabajo de Antonio Sarasa Cabezuelo “Desarrollo de un sistema de clasificación de recursos electrónicos en el ámbito de la lógica matemática” [5]:

Logic and Foundations

1.1. Logic

1.1.1. Venn Diagrams.

1.1.2. Propositional and Predicate Logic.

1.1.2.1. Propositional Logic.

1.1.2.1.1. Logical connectives.

1.1.2.1.1.1. Negation.

1.1.2.1.1.2. Conjunction.

1.1.2.1.1.3. Disjunction.

1.1.2.1.1.4. Material implication.

1.1.2.1.1.5. Biconditional.

1.1.2.1.1.6. Exclusive disjunction.

1.1.2.1.2. Formula.

1.1.2.1.2.1. Atomic formula.

1.1.2.1.2.2. Compound formulas.

1.1.2.1.2.2.1. Disjunction formula.

1.1.2.1.2.2.1.1. Disjunctive normal form.

1.1.2.1.2.2.2. Conjunction formula.

1.1.2.1.2.2.2.1. Conjunctive normal form.

1.1.2.1.2.2.3. Negation formula.

1.1.2.1.2.2.4. Material implication formula.

1.1.2.1.3. Propositional Calculus

1.1.2.1.3.1. Natural deduction.

1.1.2.1.3.1.1. Deduction

1.1.2.1.3.1.1.1. Premises.

1.1.2.1.3.1.1.2. Hypothesis.

1.1.2.1.3.1.1.3. Theorems.

1.1.2.1.3.1.1.4. Conclusion.

1.1.2.1.3.1.2. Transformation Rules.

1.1.2.1.3.1.2.1. Inference rules.

1.1.2.1.3.1.2.1.1. Introduction rules.

1.1.2.1.3.1.2.1.1.1. Reductio ad absurdum (negation introduction).

1.1.2.1.3.1.2.1.1.2. Conjunction introduction.

1.1.2.1.3.1.2.1.1.3. Disjunction introduction.

1.1.2.1.3.1.2.1.1.4. Conditional proof (conditional introduction).

1.1.2.1.3.1.2.1.1.5. Iteration.

1.1.2.1.3.1.2.1.2. Elimination rules.

1.1.2.1.3.1.2.1.2.1. Double negative elimination.

1.1.2.1.3.1.2.1.2.2. Conjunction elimination.

1.1.2.1.3.1.2.1.2.3. Disjunction elimination.

1.1.2.1.3.1.2.1.2.4. Modus ponens (conditional elimination).

1.1.2.1.3.1.2.2. Derived rules.

- 1.1.2.1.3.1.2.2.1. Hypothetical Syllogism.
- 1.1.2.1.3.1.2.2.2. Ex falso sequitur quodlibet.
- 1.1.2.1.3.1.2.2.3. Disjunctive Syllogism.
- 1.1.2.1.3.1.2.2.4. Modus Tollens.
- 1.1.2.1.3.1.2.2.5. Resolution.
- 1.1.2.1.3.1.3. Deductive equivalences.
  - 1.1.2.1.3.1.3.1. Double Negation.
  - 1.1.2.1.3.1.3.2. Transposition.
  - 1.1.2.1.3.1.3.3. Material Implication.
  - 1.1.2.1.3.1.3.4. De Morgan's Theorems.
  - 1.1.2.1.3.1.3.5. Importation-Exportation.
- 1.1.2.1.3.1.4. Proofs strategies
  - 1.1.2.1.3.1.4.1. Direct.
  - 1.1.2.1.3.1.4.2. Reductio ad absurdum.
  - 1.1.2.1.3.1.4.3. Conclusion-driven building.
  - 1.1.2.1.3.1.4.4. Premises-driven building.
- 1.1.2.1.3.2. Boolean Algebra
  - 1.1.2.1.3.2.1. Values
    - 1.1.2.1.3.2.1.1. True
    - 1.1.2.1.3.2.1.2. False
  - 1.1.2.1.3.2.2. Operations
    - 1.1.2.1.3.2.2.1. Disjunction
    - 1.1.2.1.3.2.2.2. Conjunction
  - 1.1.2.1.3.2.3. Laws
    - 1.1.2.1.3.2.3.1. Axioms
      - 1.1.2.1.3.2.3.1.1. Associativity.
      - 1.1.2.1.3.2.3.1.2. Commutativity.
      - 1.1.2.1.3.2.3.1.3. Absorption.
      - 1.1.2.1.3.2.3.1.4. Distributivity.
      - 1.1.2.1.3.2.3.1.5. Complements.
      - 1.1.2.1.3.2.3.1.6. Idempotent.
      - 1.1.2.1.3.2.3.1.7. Boundedness.
    - 1.1.2.1.3.2.3.2. Derivations.
- 1.1.2.1.3.3. Resolution Calculus Propositional.
  - 1.1.2.1.3.3.1. Clauses.
    - 1.1.2.1.3.3.1.1. Literal.
      - 1.1.2.1.3.3.1.1.1. Non-eliminable literal.
    - 1.1.2.1.3.3.1.2. Resolvent.
    - 1.1.2.1.3.3.1.3. Empty clause.
  - 1.1.2.1.3.3.2. Resolution Strategy.
    - 1.1.2.1.3.3.2.1. Proof by contradiction
      - 1.1.2.1.3.3.2.1.1. Contradiction.
      - 1.1.2.1.3.3.2.1.2. Theorem.
    - 1.1.2.1.3.3.2.2. Transformation rules.
      - 1.1.2.1.3.3.2.2.1. Resolution Rule.
      - 1.1.2.1.3.3.2.2.2. Pure Literal Rule.
      - 1.1.2.1.3.3.2.2.3. Subsumption Rule.
  - 1.1.2.1.3.3.3. Resolution techniques.
    - 1.1.2.1.3.3.3.1. Lineal Resolution.
      - 1.1.2.1.3.3.3.1.1. Troncal Clause
      - 1.1.2.1.3.3.3.1.2. Lateral Clause.
      - 1.1.2.1.3.3.3.1.3. Resolution Lineal Tree.
    - 1.1.2.1.3.3.3.2. Backtracking.
    - 1.1.2.1.3.3.3.3. Set of support strategy.
- 1.1.2.1.4. Interpretation of a truth-functional propositional calculus
  - 1.1.2.1.4.1. Interpretation.
    - 1.1.2.1.4.1.1. Atomic formula
      - 1.1.2.1.4.1.1.1. True value
      - 1.1.2.1.4.1.1.2. False value.

- 1.1.2.1.4.1.2. Calculus method
    - 1.1.2.1.4.1.2.1. Truth table.
    - 1.1.2.1.4.1.2.2. Calculus rules
      - 1.1.2.1.4.1.2.2.1. Conjunction Rule
      - 1.1.2.1.4.1.2.2.2. Disjunction Rule
      - 1.1.2.1.4.1.2.2.3. Negation Rule
      - 1.1.2.1.4.1.2.2.4. Material implication Rule.
  - 1.1.2.1.4.2. Formula interpretation
    - 1.1.2.1.4.2.1. Tautology.
    - 1.1.2.1.4.2.2. Antinomy.
    - 1.1.2.1.4.2.3. Contingency.
    - 1.1.2.1.4.2.4. Consistent.
    - 1.1.2.1.4.2.5. Semantic consequence.
    - 1.1.2.1.4.2.6. Deductive equivalence.
  - 1.1.2.1.4.3. Interpretation techniques.
    - 1.1.2.1.4.3.1. Refutation reasoning
      - 1.1.2.1.4.3.1.1. Counterexample.
    - 1.1.2.1.4.3.2. Validation reasoning
    - 1.1.2.1.4.3.3. Consistent reasoning
- 1.1.2.2. Predicate Logic
- 1.1.2.2.1. First-order predicate
    - 1.1.2.2.1.1. Term
      - 1.1.2.2.1.1.1. Proposition
      - 1.1.2.2.1.1.2. Variable
        - 1.1.2.2.1.1.2.1. Free
        - 1.1.2.2.1.1.2.2. Bounded
      - 1.1.2.2.1.1.3. Function
    - 1.1.2.2.1.2. Predicate
  - 1.1.2.2.2. Quantifiers
    - 1.1.2.2.2.1. Universal quantifier.
    - 1.1.2.2.2.2. Existential quantifier.
  - 1.1.2.2.3. Formula
    - 1.1.2.2.3.1. Atomic formula.
    - 1.1.2.2.3.2. Compound formulas.
      - 1.1.2.2.3.2.1. Disjunction formula.
      - 1.1.2.2.3.2.2. Conjunction formula.
      - 1.1.2.2.3.2.3. Negation formula.
      - 1.1.2.2.3.2.4. Material implication formula.
      - 1.1.2.2.3.2.5. Quantifier formula
        - 1.1.2.2.3.2.5.1. Existential formula.
        - 1.1.2.2.3.2.5.2. Universal formula.
        - 1.1.2.2.3.2.5.3. Free formula.
        - 1.1.2.2.3.2.5.4. Bounded formula
      - 1.1.2.2.3.2.6. Prenex normal form
  - 1.1.2.2.4. First-order predicate calculus.
    - 1.1.2.2.4.1. Natural deduction.
      - 1.1.2.2.4.1.1. Deduction
        - 1.1.2.2.4.1.1.1. Premises.
        - 1.1.2.2.4.1.1.2. Hypothesis.
        - 1.1.2.2.4.1.1.3. Theorems.
        - 1.1.2.2.4.1.1.4. Conclusion.
      - 1.1.2.2.4.1.2. Transformation Rules.
        - 1.1.2.2.4.1.2.1. Inference rules-Propositional calculus.
          - 1.1.2.2.4.1.2.1.1. Introduction rules.
            - 1.1.2.1.3.1.2.1.1.1. Reductio ad absurdum (negation introduction).
            - 1.1.2.1.3.1.2.1.1.2. Conjunction introduction.

- 1.1.2.1.3.1.2.1.1.3. Disjunction introduction.
- 1.1.2.1.3.1.2.1.1.4. Conditional proof (conditional introduction).
- 1.1.2.1.3.1.2.1.1.5. Iteration.
- 1.1.2.2.4.1.2.1.2. Elimination rules.
  - 1.1.2.1.3.1.2.1.2.1. Double negative elimination.
  - 1.1.2.1.3.1.2.1.2.2. Conjunction elimination.
  - 1.1.2.1.3.1.2.1.2.3. Disjunction elimination.
  - 1.1.2.1.3.1.2.1.2.4. Modus ponens (conditional elimination).
- 1.1.2.2.4.1.2.2. Inference rules-Predicate calculus.
  - 1.1.2.2.4.1.2.2.1. Introduction rules.
    - 1.1.2.1.3.1.2.1.1.1. Universal quantifier introduction.
    - 1.1.2.1.3.1.2.1.1.2. Existential quantifier introduction.
  - 1.1.2.2.4.1.2.2.2. Elimination rules.
    - 1.1.2.1.3.1.2.1.2.1. Universal quantifier elimination
    - 1.1.2.1.3.1.2.1.2.2. Existential quantifier elimination
- 1.1.2.2.4.1.2.3. Derived rules-Propositional calculus.
  - 1.1.2.2.4.1.2.3.1. Hypothetical Syllogism.
  - 1.1.2.2.4.1.2.3.2. Ex falso sequitur quodlibet.
  - 1.1.2.2.4.1.2.3.3. Disjunctive Syllogism.
  - 1.1.2.2.4.1.2.3.4. Modus Tollens.
  - 1.1.2.2.4.1.2.3.5. Resolution.
- 1.1.2.2.4.1.2.4. Derived rules-Predicate calculus.
  - 1.1.2.2.4.1.2.4.1. Existential quantifier distributive law over conjunction.
  - 1.1.2.2.4.1.2.4.2. Universal quantifier distributive law over disjunction.
  - 1.1.2.2.4.1.2.4.3. Existential quantifier distributive law over material implication.
  - 1.1.2.2.4.1.2.4.4. Universal quantifier distributive law over material implication.
  - 1.1.2.2.4.1.2.4.5. Universal quantifier to Existential quantifier law
- 1.1.2.2.4.1.3. Deductive equivalences-Propositional calculus
  - 1.1.2.2.4.1.3.1. Double Negation.
  - 1.1.2.2.4.1.3.2. Transposition.
  - 1.1.2.2.4.1.3.3. Material Implication.
  - 1.1.2.2.4.1.3.4. De Morgan's Theorems.
  - 1.1.2.2.4.1.3.5. Importation-Exportation.
- 1.1.2.2.4.1.4. Deductive equivalences-Predicate calculus
  - 1.1.2.2.4.1.4.1. Rewriting variables equivalence
  - 1.1.2.2.4.1.4.2. Quantifier commutative laws.
  - 1.1.2.2.4.1.4.3. Universal quantifier distributive law over conjunction.
  - 1.1.2.2.4.1.4.4. Existential quantifier distributive law over disjunction.
  - 1.1.2.2.4.1.4.5. De Morgan's Equivalences
  - 1.1.2.2.4.1.4.6. Importation-Exportation.
- 1.1.2.2.4.1.5. Proofs strategies
  - 1.1.2.2.4.1.5.1. Direct.

- 1.1.2.2.4.1.5.2.Reductio ad absurdum.
- 1.1.2.2.4.1.5.3.Conclusion-driven building.
- 1.1.2.2.4.1.5.4.Premises-driven building.
- 1.1.2.2.4.2. Resolution Calculus Predicate
  - 1.1.2.2.4.2.1. Clauses.
    - 1.1.2.2.4.2.1.1. Literal.
      - 1.1.2.2.4.2.1.1.1. Non-eliminable literal.
    - 1.1.2.2.4.2.1.2.Resolvent.
    - 1.1.2.2.4.2.1.3.Empty clause.
  - 1.1.2.2.4.2.2. Skolem normal form.
  - 1.1.2.2.4.2.3. Resolution Strategy.
    - 1.1.2.2.4.2.3.1.Proof by contradiction
      - 1.1.2.2.4.2.3.1.1. Contradiction.
      - 1.1.2.2.4.2.3.1.2. Theorem.
    - 1.1.2.2.4.2.3.2.Transformation rules.
      - 1.1.2.2.4.2.3.2.1. Resolution Rule.
      - 1.1.2.2.4.2.3.2.2. Unification Rule.
      - 1.1.2.2.4.2.3.2.3. Pure Literal Rule.
      - 1.1.2.2.4.2.3.2.4. Subsuntion Rule.
    - 1.1.2.2.4.2.4.Resolution techniques.
      - 1.1.2.2.4.2.4.1. Lineal Resolution.
        - 1.1.2.2.4.2.4.1.1. Troncal Clause
        - 1.1.2.2.4.2.4.1.2. Lateral Clause.
        - 1.1.2.2.4.2.4.1.3. Resolution Lineal Tree.
      - 1.1.2.2.4.2.4.2. Backtracking.
      - 1.1.2.2.4.2.4.3. Set of support strategy.
- 1.1.2.2.5. Interpretation of a truth-functional predicate calculus
  - 1.1.2.2.5.1. Interpretation.
    - 1.1.2.2.5.1.1. Atomic formula
      - 1.1.2.2.5.1.1.1.True value
      - 1.1.2.2.5.1.1.2. False value.
    - 1.1.2.2.5.1.2.Calculus method
      - 1.1.2.2.5.1.2.1. Variable Domain.
        - 1.1.2.2.5.1.2.2. Interpretation for predicate.
        - 1.1.2.2.5.1.2.3. Transformation rules: Predicate to proposition.
          - 1.1.2.2.5.1.2.3.1. Universal formula to proposition.
          - 1.1.2.2.5.1.2.3.2. Existential formula to proposition.
        - 1.1.2.2.5.1.2.4. Truth table.
        - 1.1.2.2.5.1.2.5.Calculus rules- Proposition
          - 1.1.2.2.5.1.2.5.1. Conjunction Rule
          - 1.1.2.2.5.1.2.5.2. Disjunction Rule
          - 1.1.2.2.5.1.2.5.3. Negation Rule
          - 1.1.2.2.5.1.2.5.4. Material implication Rule.
  - 1.1.2.2.5.2. Formula interpretation
    - 1.1.2.2.5.2.1. Tautology.
    - 1.1.2.2.5.2.2.Antinomy.
    - 1.1.2.2.5.2.3.Contingency.

- 1.1.2.2.5.2.4. Consistent.
- 1.1.2.2.5.2.5. Semantic consequence.
- 1.1.2.2.5.2.6. Deductive equivalence.
- 1.1.2.2.5.3. Interpretation techniques.
  - 1.1.2.2.5.3.1. Refutation reasoning
    - 1.1.2.2.5.3.1.1. Counterexample.
  - 1.1.2.2.5.3.2. Validation reasoning
  - 1.1.2.2.5.3.3. Consistent reasoning

## 1.2. Set Theory

- 1.2.1. Sets and Set Operations
  - 1.2.1.1. Element
  - 1.2.1.2. Empty set
  - 1.2.1.3. Universal set
  - 1.2.1.4. Complementary set
  - 1.2.1.5. Union
  - 1.2.1.6. Intersection
  - 1.2.1.7. Difference
  - 1.2.1.8. Disjoint sets
  - 1.2.1.9. Cartesian product
    - 1.2.1.9.1. Ordered pair
    - 1.2.1.9.2. n-tuple
  - 1.2.1.10. Power set
- 1.2.2. Relations and Foundations
  - 1.2.2.1. Membership relation
  - 1.2.2.2. Binary relation
    - 1.2.2.2.1. Reflexive
    - 1.2.2.2.2. Antireflexive
    - 1.2.2.2.3. Symmetric
    - 1.2.2.2.4. Antysymmetric
    - 1.2.2.2.5. Transitive
  - 1.2.2.3. Equivalence relation
    - 1.2.2.3.1. Equivalence class
    - 1.2.2.3.2. Partition
    - 1.2.2.3.3. Quotient set
  - 1.2.2.4. Congruence relation
  - 1.2.2.5. Ordering relation
    - 1.2.2.5.1. Stric ordering
    - 1.2.2.5.2. Partial ordering
    - 1.2.2.5.3. Total ordering
  - 1.2.2.6. Relation operations
    - 1.2.2.6.1. Selection
    - 1.2.2.6.2. Projection
    - 1.2.2.6.3. Combination
- 1.2.3. Cardinality
  - 1.2.3.1. Limit Cardinalities 2
- 1.2.4. Axiom of Choice
- 1.2.5. Correspondence
- 1.2.6. Functions
  - 1.2.6.1. Domain
  - 1.2.6.2. Range
  - 1.2.6.3. Image
  - 1.2.6.4. Identity function
  - 1.2.6.5. Injective
  - 1.2.6.6. Surjective / Onto
  - 1.2.6.7. Bijective
  - 1.2.6.8. Inverse function Partial function
  - 1.2.6.9. Total function

## 1.3. Computability and Decidability

## 1.4. Model Theory

El XML de la taxonomía debe de tener la siguiente estructura para que sea admitido por el sistema, independientemente del campo de la material, ya que éste admite la gestión de varias taxonomías:

```
<?xml version="1.0" encoding="UTF-8"?>
<vdex ordeSignificant="true" profileType="hierarchical Token Terms"
language="en"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imsvdexv1p0
imsvdex_v1p0.xsd"
xmlns="http://www.imsglobal.org/xsd/imsvdex_v1p0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<vocabName>
  <langstring>"Core" Subject Taxonomy for Mathematical Sciences
Education
  </langstring>
</vocabName>
<term>
  <termIdentifier> 1 </termIdentifier>
  <caption>
    <langstring>Numbers and Computation </langstring>
  </caption>
  <term>
    <termIdentifier> 1.1 </termIdentifier>
    <caption>
      <langstring>Number Concepts</langstring>
    </caption>
    <term>
      <termIdentifier> 1.1.1 </termIdentifier>
      <caption>
        <langstring>Natural</langstring>
      </caption>
    </term>
  </term>
</term>
</term>
```

[...]  
Continúa...



## Anexo 3: Normas de Accesibilidad Web

### *Accesibilidad Web para Discapacitados*

Los tipos de discapacidades que existen y pueden dificultar el uso del sitio web son:

- **Discapacidades visuales:** Las personas que padecen este tipo de discapacidades necesitan ver con claridad el texto o las imágenes de la pantalla para poder llevar a cabo tareas que requieren la coordinación de manos y ojos, como mover correctamente el ratón. **El tamaño y el color del texto pueden marcar una gran diferencia en materia de legibilidad para personas con visión reducida.**
- **Discapacidades de movimiento:** El control reducido de los músculos o su debilidad puede dificultar el uso de dispositivos de teclado y ratón estándar. Por ejemplo, algunas personas no pueden presionar dos teclas simultáneamente, mientras que otras tienden a presionar varias teclas o a presionar una tecla varias veces de forma involuntaria cuando quieren presionarlas y soltarlas. **Las personas que sólo pueden usar una mano tienen dificultades con algunas tareas del teclado o el ratón.**
- **Discapacidades auditivas:** Las personas con dificultades auditivas quizás puedan oír algunos sonidos pero es posible que no puedan distinguir las palabras. Otras quizás no puedan oír ningún sonido. **Al no poder oír las advertencias del equipo, como los pitidos y mensajes de voz, pueden tener problemas.**
- **Discapacidades cognitivas y de lenguaje:** Las discapacidades cognitivas y de lenguaje abarcan tanto la dislexia como dificultades para recordar, resolver problemas o percibir información sensorial o incluso problemas para comprender y utilizar el lenguaje. Para personas con estas dificultades, el uso de los equipos puede verse dificultado por **pantallas complejas o incoherentes** o por la elección de las palabras.
- **Epilepsia:** Ciertos patrones de luz o sonido pueden provocar ataques epilépticos en algunas personas susceptibles de padecerlos.
- **Discapacidades relacionadas con la edad:** El problema más común al envejecer es el deterioro natural de la vista. A los 65 años, la mayoría de las personas ha perdido parte de su capacidad para enfocar, resolver imágenes, distinguir colores y adaptarse a los cambios de luz. Con la edad, también suelen experimentarse otros efectos degenerativos. Así, son habituales diversos grados de pérdida de audición, al igual que las dificultades de coordinación motora, a menudo debidas a la artritis o al desgaste de las articulaciones.

## ***Estándares técnicos recomendados***

Las normas a cumplir para conseguir la **Accesibilidad de un sitio web** están separadas en tres áreas a las que se les asigna diferente nivel de Prioridad:

- **Prioridad 1:** Un desarrollador de contenidos de páginas Web tiene que satisfacer este punto de verificación, porque de lo contrario, uno o más grupos de usuarios encontrarán imposible acceder a la información del documento. Satisfacer este punto de verificación es un requerimiento básico para que algunos grupos puedan usar estos documentos Web.
- **Prioridad 2:** Un desarrollador de contenidos de páginas Web debe satisfacer este punto de verificación, porque de lo contrario, uno o más grupos encontrarán dificultades en el acceso a la información del documento. Satisfacer este punto de verificación eliminará importantes barreras de acceso a los documentos Web.
- **Prioridad 3:** Un desarrollador de contenidos de páginas Web puede satisfacer este punto de verificación., porque de lo contrario, uno o más grupos de usuarios encontrarán alguna dificultad para acceder a la información del documento. Satisfacer este punto de verificación mejorará la accesibilidad de los documentos Web.

Como se ve en la descripción anterior, cada una de las prioridades lleva asociado un tipo de obligación, siendo la primera la más importante de todas.

### Puntos de verificación para cumplir la Prioridad 1:

- Proporcione un texto equivalente para todo elemento no textual (por ejemplo, a través de "alt"). Esto incluye: imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (por ejemplo, GIFs animados), "applets" y objetos programados, "ASCII art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (utilizados con o sin interacción), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos.
- Asegure que toda la información transmitida a través de los colores también esté disponible sin color, por ejemplo mediante el contexto o por marcadores.
- Identifique claramente los cambios en el idioma original del texto del documento y en cualquier texto equivalente (por ejemplo, leyendas).
- Organice el documento de forma que pueda ser leído sin hoja de estilo. Por ejemplo, cuando un documento HTML es interpretado sin asociarlo a una hoja de estilo CSS, tiene que ser posible leerlo y entenderlo.

- Asegure que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico.
- Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar parpadeo en la pantalla.
- Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio.

Si utiliza imágenes y mapas de imagen (Prioridad 1)...

- Proporcione vínculos de texto redundantes con cada zona activa de un mapa de imagen del servidor.
- Proporcione mapas de imagen controlados por el cliente en lugar de por el servidor, excepto donde las zonas sensibles no puedan ser definidas con una forma geométrica.

Y si utiliza tablas (Prioridad 1)...

- En las tablas de datos, identifique los encabezamientos de fila y columna.
- Para las tablas de datos que tienen dos o más niveles lógicos de encabezamientos de fila o columna, utilice marcadores para asociar las celdas de encabezamiento y las celdas de datos.

Y si utiliza marcos ("frames") (Prioridad 1)...

- Titule cada marco para facilitar la identificación y navegación de los mismos.

Y si utiliza "applets" y "scripts" (Prioridad 1)...

- Asegure que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, applets u otros objetos de programación. Si esto no es posible, proporcione información equivalente en una página alternativa accesible

Y si utiliza multimedia (Prioridad 1)...

- Hasta que las aplicaciones de usuario puedan leer automáticamente el texto equivalente de la banda visual, proporcione una descripción auditiva de la información importante de la banda visual de una presentación multimedia.
- Para toda presentación multimedia tempo dependiente (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda de visual) con la presentación.

Y si todo lo demás falla (Prioridad 1)...

- Si, después de los mayores esfuerzos, no puede crear una página accesible, proporcione un vínculo a una página alternativa parecida que use tecnologías W3C y sí que sea accesible.

Para cumplir lo indicado en Prioridad 2, las tareas son las siguientes:

- Asegure que las combinaciones de los colores de fondo y primer plano tengan el suficiente contraste para que sean percibidas por personas con deficiencias de percepción de color o por pantallas en blanco y negro [Prioridad 2 para las imágenes. Prioridad 3 para los textos].
- Cuando exista un marcador apropiado, use marcadores en vez de imágenes para transmitir la información.
- Cree documentos que estén validados por las gramáticas formales publicadas.
- Utilice hojas de estilo para controlar la diagramación y la presentación.
- Utilice unidades relativas en lugar de absolutas al especificar los valores en los atributos de los marcadores de lenguaje y en los valores de las propiedades de las hojas de estilo.
- Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación.
- Marque las listas y los puntos de las listas correctamente.
- Marque las citas. No utilice el marcador de citas para efectos de formato tales como sangrías.
- Asegure que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa.
- Hasta que las aplicaciones de usuario permitan controlarlo, evite el parpadeo del contenido (por ejemplo, cambio de presentación en periodos regulares, así como el encendido y apagado).
- Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica.
- Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener el redireccionamiento automático, no utilice marcadores para redirigir las páginas automáticamente. En su lugar, configure el servidor para que ejecute esta posibilidad.
- Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario.
- Utilice tecnologías W3C cuando estén disponibles y sean apropiadas para la tarea, y use las últimas versiones cuando sean soportadas.

- Divida los bloques largos de información en grupos más manejables cuando sea natural y apropiado.
- Identifique claramente el objetivo de cada vínculo.
- Proporcione meta datos para añadir información semántica a las páginas y sitios.
- Proporcione información sobre la maquetación general de un sitio (por ejemplo, mapa del sitio o tabla de contenidos).
- Utilice los mecanismos de navegación de forma coherente.

Si utiliza tablas (Prioridad 2)...

- No utilice tablas para representar diagramas, a menos que la tabla tenga sentido cuando se alinee.
- Si se utiliza una tabla para representar un diagrama, no utilice marcadores estructurales para realizar un formateo visual.

Y si utiliza marcos ("frames") (Prioridad 2)...

- Describa el propósito de los marcos y cómo éstos se relacionan entre sí, si no resulta obvio solamente con el título del marco.

Y si utiliza formularios (Prioridad 2)...

- Hasta que las aplicaciones de usuario soporten explícitamente la asociación entre control de formulario y etiqueta, para todos los controles de formularios con etiquetas asociadas implícitamente, asegure que la etiqueta está colocada adecuadamente.
- Asocie explícitamente las etiquetas con sus controles.

Y si utiliza "applets" y "scripts" (Prioridad 2)...

- Para los scripts y applets, asegure que los manipuladores de eventos sean entradas independientes del dispositivo.
- Hasta que las aplicaciones de usuario permitan congelar el movimiento de los contenidos, evite los movimientos en las páginas.
- Haga los elementos de programación, tales como scripts y applets, directamente accesibles o compatibles con las ayudas técnicas [Prioridad 1 si la funcionalidad es importante y no se presenta en otro lugar; de otra manera, Prioridad 2].
- Asegure que cualquier elemento que tiene su propia interfaz pueda manejarse de forma independiente del dispositivo.
- Para scripts, especifique manipuladores de eventos lógicos mejor que manipuladores de eventos dependientes de dispositivos.

Para cumplir lo indicado en Prioridad 3, las tareas son las siguientes:

- Especifique la expansión de cada abreviatura y acrónimo en el documento cuando aparezcan por primera vez.
- Identifique el idioma principal de un documento.
- Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos.
- Proporcione atajos de teclado para los vínculos más importantes, los controles de formulario y los grupos de controles de formulario.
- Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten claramente los hipervínculos contiguos, incluya caracteres imprimibles rodeados de espacios, entre cada dos vínculos contiguos.
- Proporcione la información de modo que los usuarios puedan recibir los documentos según sus preferencias (por ejemplo, idioma, tipo de contenido, etc.).
- Proporcione barras de navegación para destacar y dar acceso al mecanismo de navegación.
- Agrupe los vínculos relacionados, identifique el grupo (para las aplicaciones de usuario) y, hasta que las aplicaciones de usuario lo hagan, proporcione una manera de evitar el grupo.
- Si proporciona funciones de búsqueda, permita diferentes tipos de búsquedas para diversos niveles de habilidad y preferencias.
- Proporcione información sobre las colecciones de documentos (por ejemplo, los documentos que comprendan múltiples páginas).
- Proporcione un medio para saltar sobre un ASCII art de varias líneas.
- Complemente el texto con presentaciones gráficas o auditivas cuando ello facilite la comprensión de la página.
- Cree un estilo de presentación que sea coherente para todas las páginas.

Y si utiliza imágenes o mapas de imagen (Prioridad 3):

- Hasta que las aplicaciones de usuario interpreten el texto equivalente para los vínculos de los mapas de imagen de cliente, proporcione vínculos de texto redundantes para cada zona activa del mapa de imagen de cliente.

