Sistemas Informáticos
Curso 2004-2005

Creación, Gestión y Uso de Objetos de Aprendizaje en la Web

Juan José Calvo Diaz De Neira
Alberto Espadero Guillermo
David Sánchez LLorente

Dirigido por:
Prof. Alfredo Fernández-Valmayor Crespo
Dpto. Sistemas Informáticos y Programación

Facultad de Informática
Universidad Complutense de Madrid

Índice

AGRADECIMIENTOS

Nos es grato haber contado con la colaboración de nuestro director de proyecto Alfredo Fernández-Valmayor y de la diseñadora gráfica Sara Olmos por el tiempo que nos han dedicado y por su apoyo en todos estos meses de trabajo.

Resumen

El proyecto “Creación, Gestión y Uso de objetos de Aprendizaje en un Entorno Web” se ha desarrollado en la asignatura Sistemas Informáticos. Tiene como objetivo la puesta en producción de la versión 1.0 del Museo Virtual García Santesmases de la Facultad de Informática de la Universidad Complutense de Madrid.

Este proyecto se centra en la construcción de una herramienta que sirva para la creación y gestión de recursos educativos, *Objetos de Aprendizaje*, modulares que puedan ser utilizados por diversos sistemas de enseñanza. Los *Objetos de Aprendizaje* creados con la herramienta a la que se refiere este proyecto están basados en los objetos del museo de informática García Santesmases, por lo que también se pueden utilizar como elementos de un *museo virtual*.

Abstract

The proyect called “Creation, Management and Use of Learning Objects in the Web” has been developed into the subject “Sistemas Informáticos”. The main target is the setting in production of the 1.0 version of the “Virtual García Santesmases Museum” located at the School of Computer Science of the Complutense University of Madrid.

This project focuses in the construction of a tool to be useful in the creation and management of educational resources, *Learning Objects*, modular Objects that they may be used by various tuitional systems. The *Learning Objects* created with the tool that this project refers to itself are based in the objects of García Santesmases Museum, so that also they can utilize themselves like elements of a virtual museum

Palabras Clave

E-Learning; Museos Virtuales; Objetos de Aprendizaje; XML; IMS; LOM

1 Introducción

Este documento proporciona una visión general del proyecto "Creación, Gestión y Uso de Objetos de Aprendizaje en un entorno Web" que se ha desarrollado en la asignatura de Sistemas Informáticos durante el curso 2004/2005.

Este proyecto se puede enmarcar dentro de los proyectos e-learning, pero a diferencia de otros proyectos educativos éste no se centra en la construcción de herramientas informáticas para la creación de unidades didácticas, sino en la construcción de una herramienta que sirva para crear y gestionar recursos educativos, *Objetos de Aprendizaje*, modulares y que puedan ser utilizados por diversos sistemas de enseñanza. Los *Objetos de Aprendizaje* creados con la herramienta a la que se refiere este proyecto están basados en los objetos del museo de informática García Santesmases, por lo que también se pueden utilizar como elementos de un *museo virtual*.

El museo de Informática García Santesmases (MIGS) se encuentra situado en los pasillos de la Facultad de Informática de la Universidad Complutense de Madrid, conformando un "paseo histórico" por cinco décadas de Informática. Se denomina así en memoria del Profesor García Santesmases, catedrático de esta Universidad que fue pionero en la investigación y docencia de la Informática en España.

En él se exponen máquinas desarrolladas en la UCM entre los años 1950 y 1975, así como computadoras comerciales que desde 1968 estuvieron en el Centro de Cálculo de esta Universidad y equipos donados por Departamentos, particulares y otras entidades. Las piezas, organizadas cronológicamente por categorías, se complementan con paneles explicativos que muestran sus características y modo de funcionamiento, todo ello con el propósito de ofrecer una visión didáctica y global de la evolución de la informática en los últimos cincuenta años. El museo virtual García Santesmases contiene, además de las piezas expuestas en el museo real, más piezas y documentos que por diferentes razones no pueden ser mostradas en el situado físicamente en la facultad de informática.

Este trabajo tiene como objetivo la puesta en producción del prototipo del museo virtual García Santesmases ya existente completándolo con nuevas funcionalidades. Sin embargo, no se trata de un trabajo ya concluido, sino que es una etapa más de un proyecto de larga duración. Los objetivos que debía cumplir la aplicación informática consistían en permitir la creación de objetos de aprendizaje tomando como base los objetos de un museo real. Uno de los requisitos importantes de esta aplicación es la transportabilidad de los objetos de aprendizaje creados, de tal manera que éstos pueden ser reutilizados dentro de otros entornos y sistemas educativos así como ser importados de otras aplicaciones siempre que cumplan un cierto estándar.

El resultado inmediato de este proyecto ha sido la reingeniería de la versión 0.0 del Museo de Informática García Santesmases (MIGS) así como su ingeniería y adaptación a los requisitos solicitados por la dirección de la Facultad de Informática de la Universidad Complutense de Madrid y por los usuarios del museo, inicialmente profesores y becarios encargados de catalogar y documentar los objetos del museo. Esta aplicación permite que el usuario investigador pueda construir objetos de aprendizaje basados en los objetos reales del museo. Posteriormente el usuario, alumno o visitante, podrá visitar a través de un entorno Web los objetos del museo. El museo virtual permite acceder además a ciertos objetos que por diversas razones no están expuestos,

tales como documentos y fotografías. Otro requisito importante de la aplicación es la modularidad, que permite que a partir de los objetos primarios del museo, expuestos y no expuestos, se puedan construir otros objetos que combinen objetos anteriores mediante enlaces a los mismos. Estos objetos compuestos facilitan que el alumno o visitante pueda establecer relaciones significativas entre los objetos del museo y estudiar y comprender el contexto en que estos objetos fueron creados.

2 Estandares de la Enseñanza en Línea (E-Learning)

2.1 Utilización de estándares en el aprendizaje en línea

Existen especificaciones y estándares para el aprendizaje en línea, pero ¿realmente son necesarias?: realizando un estudio del panorama actual, existen decenas de LMS (sitemas de gestión de aprendizaje) y de VLE (entornos de aprendizaje virtual), para poder aplicar según cada necesidad. Algunos de ellos son de código abierto como Moodle, un CMS(sistema gestor de cursos) con facilidades y servicios para el aprendizaje colaborativo, Reload, editor de Scorm y ciertas especificaciones de IMS ó como EML. Pero con ninguno de ellos se puede intercambiar contenidos y estructuras de aprendizaje. Por esto son necesarios unas especificaciones y estándares para los sistemas de aprendizaje virtual.

Una especificación sobre aprendizaje en línea permite modelar un elemento o una etapa del proceso educativo, trabajar con él y mantener el nuevo material funcionando exactamente igual, independientemente de la plataforma que se utilice. Es decir, puede ser migrado automáticamente y el contenido y estructura del curso son independientes de la plataforma de ejecución. Además se escribe en código abierto (interpretable y modificable, por tanto) y es gratis.

Un estándar es una tecnología, formato o método, reconocido nacional o internacionalmente, documentado en detalle y ratificado por una autoridad competente en su campo, como ISO ó IEEE. Por el contrario, una especificación es el paso previo, creado por alguna compañía u organismo, que no ha sido ratificado todavía por ninguna autoridad, y que suele usarse de manea provisional pero suficientemente respaldada. En esta etapa se encuentran la mayoría de las especificaciones para la enseñanza en línea.

2.2 Herramientas de Referencia

El desarrollo de este proyecto se ha basado fundamentalmente en el concepto de *Objeto de Aprendizaje* (learning objects) que ha sido promovido por iniciativas como IEEE/LTSC (Learning Technology Standards Comittee) [1], IMS [2], ADL (Advanced Distributed Learning). SCORM (Sharable Content Object Reference Model) [3], promovida por ADL, es una especificación para la construcción de objetos de aprendizaje complejos a partir de otros más sencillos.

Según L. Mortimer [6] algunos de los aspectos que caracterizan a un objeto de aprendizaje son:

- *Contenido*: contenido y actividades deben soportar un aprendizaje objetivo, y su evaluación debe de estar orientada a conseguir dicha objetividad.
- *Tamaño*: Realizar las actividades implicadas por un objeto de aprendizaje no debe llevar mucho tiempo.
- *Contexto y capacidades*: Un objeto de aprendizaje debe poder existir de forma independiente y debe poder ser utilizado por cualquier participante (estudiante) que tenga determinadas capacidades y en el momento que lo necesite.
- *Etiquetado y almacenado*: El contenido se describe mediante un conjunto de etiquetas normalizadas (metadatos).

- *Construcción incremental*: Los objetos de aprendizaje complejos (con una estructura bien definida) se construyen utilizando otros objetos de aprendizaje previamente construidos partiendo de objetos de aprendizaje más básicos (hasta llegar a los objetos atómicos).
- *Interdependencia*: Un objeto de aprendizaje se considera que está formado por tres componentes interdependientes: el objeto de aprendizaje en sí mismo, los meta-datos (la forma estandarizada de describir su contenido) y un componente de gestión del aprendizaje (LMS o *Learning Management System*) que almacena, localiza y entrega contenidos.

Uno de los conceptos más importantes sobre el que se asienta este proyecto es el concepto de *Objeto Virtual* (OV), un tipo específico de *Objeto de Aprendizaje* (Learning Object). Un *Objeto Virtual* puede ser definido como *"un objeto digital que sirve para agrupar con fines educativos toda la información relacionada con un determinado objeto real"* [4]. Así pues la estructura de datos de un *Objeto Virtual* está especificada como:

- **Datos**: Permiten organizar y describir de forma extensible las características del objeto.
- **Recursos**: Constituyen un conjunto de elementos informativos asociados al objeto. Dichos recursos pueden ser considerados de tres clases:
  - *Propios*: Archivos digitales asociados de forma primaria a un objeto virtual.
  - *Ajenos*: Archivos digitales asociados de forma primaria a otros objetos del almacén pero que guardan algún tipo de relación con el objeto actual.
  - *Otros Objetos Virtuales*: son considerados de forma unitaria y asociados al objeto virtual estudiado mediante una relación describiendo las propiedades que pueden incorporarse al objeto.
- **Metadatos**: Describen el contenido, calidad, condiciones y otras características de los datos, permitiendo así a una persona ubicar y entender los datos. Cada uno de estos objetos puede ser descrito mediante un modelo de metadatos, y en este caso están basados en la propuesta de LOM (Learning Object Metadata) [5], cuyo modelo permite disponer de una forma estándar de clasificación de la información y establecer las posibles relaciones existentes entre los diferentes *Objetos Virtuales*.

Además de las características de los objetos de aprendizaje ya mencionadas, también se ha dado gran importancia durante el desarrollo del proyecto a los estándares educativos que a continuación se introducen.

SCORM [3], iniciativa de ADL, ha sido la principal referencia para este proyecto. Otras iniciativas que también nos han servido de referencia han sido: ARIADNE, IEEE/LTSC e IMS. En SCORM [3] se define un "modelo de agregación de contenido" (content agregation model), el cual debe de ser pedagógicamente neutro. El modelo de agregación de contenido propuesto por SCORM [3] está formado básicamente por tres elementos: el modelo de contenido, los metadatos, y el modelo de empaquetamiento. Así mismo, el modelo de contenido se compone a su vez de tres

elementos: assets (los contenidos más básicos), SCOs (Sharable Content Object) conjunto de assets que pueden ejecutarse en el entorno de enseñanza del sistema y finalmente las estructuras de agregación (content agregation) las cuales son capaces de organizar diferentes recursos de aprendizaje hasta formar una unidad didáctica coherente.

La propuesta de LOM (Learning Object Metadata) se ha consolidado como la principal referencia para describir mediante metadatos los recursos educativos digitales y hacerlos disponibles a través de la Web. Así, el modelo de metadatos de LOM, implica que la información referente a un objeto virtual se agrupe en categorías. El esquema básico está formado por nueve categorías:

- **General**: Engloba las características independientes del contexto además de descriptores del recurso.
- **Ciclo de vida**: Características referentes al ciclo de vida del recurso.
- **Meta-metainformación**: Aspectos de la propia descripción.
- **Técnica**: Aspectos técnicos del recurso.
- **Uso educativo**: Características educativas o pedagógicas de recurso.
- **Derechos**: Condiciones de uso del recurso.
- **Relación**: Relaciones del recurso con otro recurso.
- **Observaciones**: Permite comentarios sobre el uso del recurso.
- **Clasificación**: Características del recurso según lo describen diferentes catálogos.

Los metadatos usados en MIGS se han centrado en este estándar tomando como referencia las categorías *General*, *Ciclo de Vida* y *Clasificación*.

Por otro lado, existen diferentes grupos como Dublín Core [7] y ARIADNE [8] que han ajustado sus esquemas de metadatos a LOM, pero con menos nivel de detalle.

La aproximación que utilizamos para desarrollar este proyecto tiene como base los estándares antes descritos y el concepto de *Objeto Virtual*. El dominio de este proyecto son los objetos que se encuentran en el museo de informática García Santesmases de la Facultad de Informática de la Universidad Complutense de Madrid. Es decir, lo que buscamos es crear *Objetos Virtuales* a partir de los objetos disponibles en dicho museo, así como las posibles relaciones que se puedan establecer entre ellos.

Cada objeto virtual se describe mediante un modelo de metadatos basado en el estándar LOM. Para llevar a cabo este proceso de virtualización se ha añadido dentro del museo virtual un acceso a una herramienta que permite crear objetos virtuales según los estándares antes definidos, y que también permite modificar objetos virtuales que hayan sido creados con anterioridad, y que se encuentran almacenados en el servidor en una base de datos relacional, en la cual la entidad fundamental es el objeto virtual.

El MIGS tiene como requisito importante que la información puede ser accedida desde cualquier computador que tenga acceso a la red y disponga de un navegador web. Para diseñar este entorno Web se han usado principalmente las siguientes tecnologías:

- PHP, HTML, JavaScript para crear un entorno Web dinámico.
- XML y DTDs para construir los manifiestos de los objetos virtuales.

2.3 IMS Content Packaging (IMS-CP)

Es la especificación seguida para el empaquetamiento de objetos virtuales en el MIGS, con este empaquetamiento son subidos los nuevos objetos al museo y bajados para ser exportados a otros sistemas compatibles.

El IMS-CP describe las estructuras de datos que son usadas para proporcionar la interoperabilidad de contenidos basados en internet con herramientas de creación de contenidos, sistemas de gestión de aprendizaje (LMS), y entornos de ejecución.

El objetivo de IMS-CP es definir un conjunto estandarizado de estructuras que puedan ser usadas para intercambiar contenidos. Estas estructuras proporcionan la base para las uniones de datos estandarizadas que permiten crear aplicaciones que se comuniquen con herramientas de creación de contenidos, sistemas de gestión de aprendizaje (LMS), y entornos de ejecución que han sido implementados por separado por distintos desarrolladores de software.

El alcance de la especificación IMS CP está centrada en la definición de la interoperabilidad entre sistemas que importan, exportan, añaden y borran paquetes de contenidos.

El Modelo conceptual se especifica de la siguiente manera:

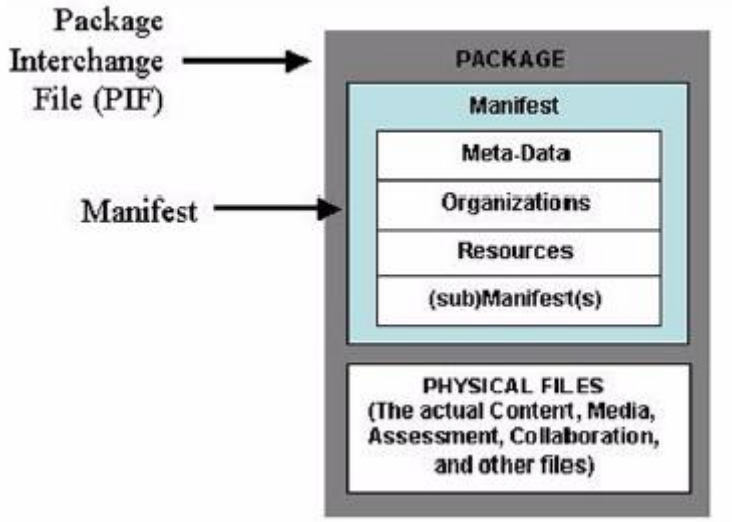

Figura 1a: Modelo IMS-CP.

- **Archivo de Intercambio de Paquetes (PIF)**: Es un archivo (.zip, .jar …), que incluye en su capa más alta, un archivo tipo "manifiesto" llamado "imsmanifest.xml" y otros archivos físicos identificados por el manifiesto. Un PIF es un conciso formato de entrega Web y medio de transportar información relacionada y estructurada.
- **Paquete**: directorio lógico, que incluye un archivo XML, en el cual, cualquier documento XML de control es directamente referenciado (como un archivo DTD ó XSD), y contiene los recursos reales físicos. Los recursos físicos pueden ser organizados en subdirectorios:
    - *Manifiesto de alto nivel*: un elemento obligatorio XML que describe el Paquete. Puede contener opcionalmente

submanifiestos. Cada manifiesto contiene las secciones siguientes:

- Meta datos: un elemento XML que describe un manifiesto en totalidad.
- Organizaciones: un elemento XML que describe cero, una, o múltiples organizaciones del contenido dentro de un manifiesto.
- Recursos: un elemento XML que contiene referencias a todos los recursos y elementos necesitado para un manifiesto, incluyendo meta-datos que describen los recursos, y referencias a cualquier archivo externo;
- Sub-manifiestos: uno o varios, opcional

o *Archivos de recursos*: estos son los elementos reales, archivos de texto, gráficos, y otros recursos descritos por el manifiesto. Los recursos físicos pueden ser organizados en subdirectorios.

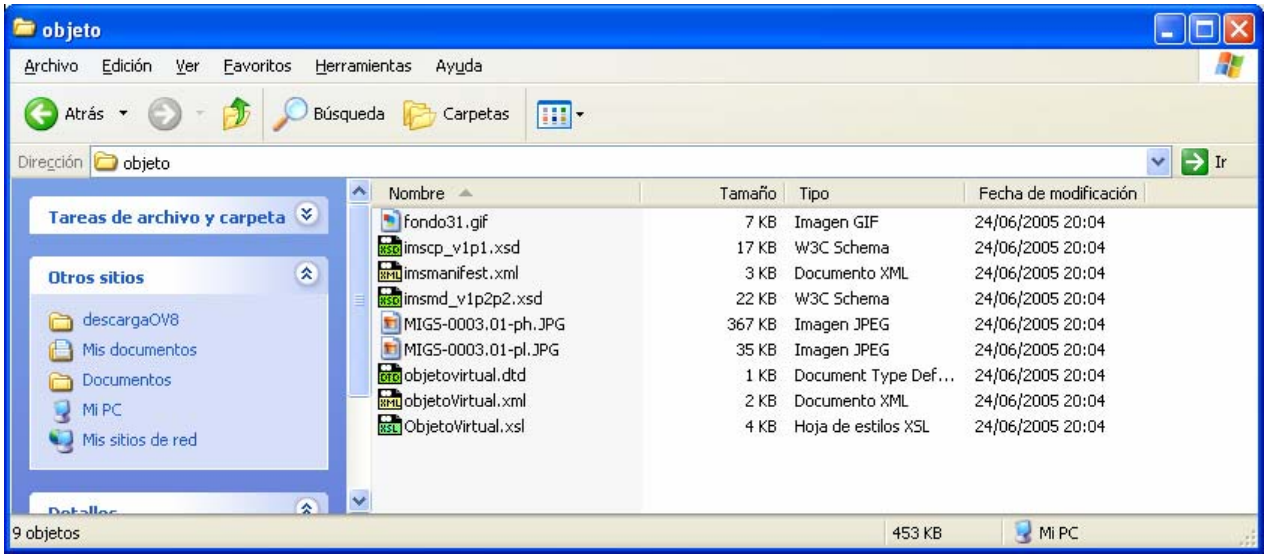

Figura 1b: Documentos del empaquetado de un Objetos virtual en MIGS.

3 Antecedentes de MIGS

El desarrollo de este proyecto está basado en otros desarrollos de museos virtuales: Chasqui y MIGS 0.0, en los que también ha colaborado la Facultad de Informática de la Universidad Complutense de Madrid. En ellos existe un repositorio de objetos de aprendizaje en una base de datos el cual está pensado para acceder a él a través de una interfaz Web o a través de servicios Web (Web services).

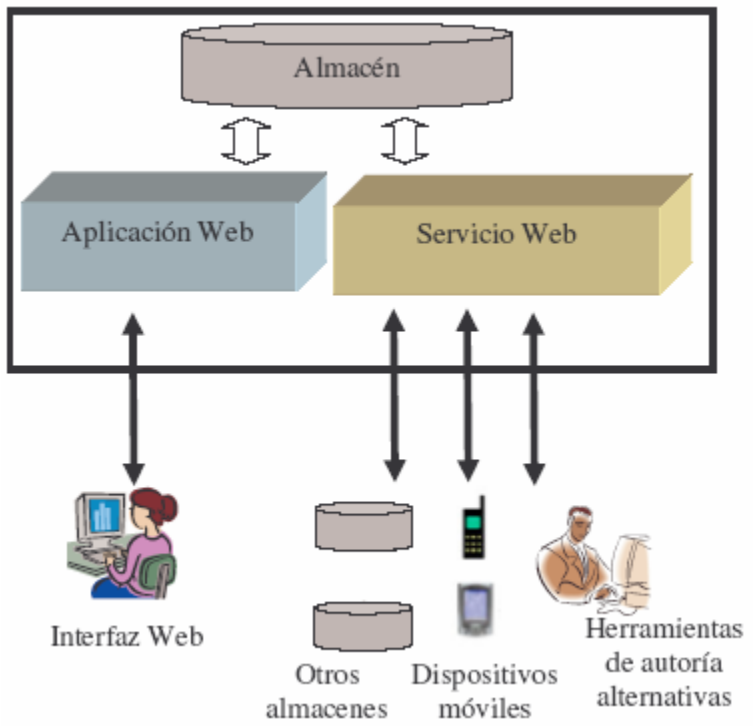

Figura 2: Arquitectura Global de los museos.

En este proyecto nos centraremos en la parte de la aplicación Web. Tanto en Chasqui como en MIGS 0.0 la metodología seguida para el desarrollo ha sido como se especifica a continuación:

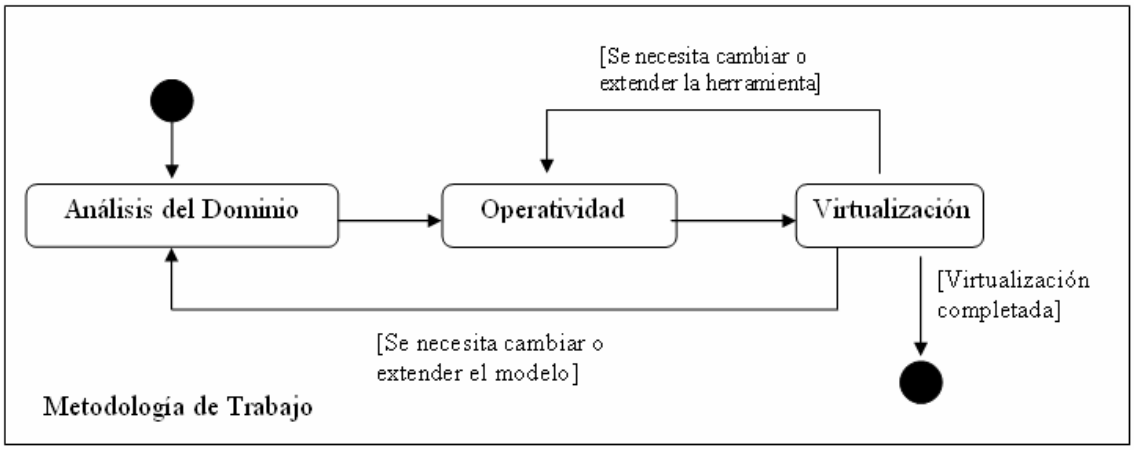

Figura 3: Metodología de Trabajo.

Esta metodología propone una estrategia iterativa e incremental de la formulación y refinamiento del modelo de *Objeto de Aprendizaje* y de la herramienta de autoría en base a la experiencia. La secuenciación de actividades mostrada en la figura

refleja dicha estrategia. De esta forma, en lugar de llevar a cabo un análisis del dominio exhaustivo, seguido de una operacionalización y virtualización igualmente exhaustivas, estas actividades se solapan y realimentan en el tiempo.

La arquitectura software que se ha usado sigue el patrón denominado Modelo-Vista-Controlador (MVC), en el que se lleva a cabo una separación entre el modelo de datos, la interfaz del usuario y la lógica de control. De esta forma un componente puede ser modificado sin afectar excesivamente a los demás.

El modelo representa los datos y las operaciones que permiten el acceso y la modificación de los datos. Normalmente el modelo notifica a la vista los cambios que se producen en los datos.

La vista accede a los datos que contiene el modelo y especifica la forma en que éstos deben ser mostrados. La vista es actualizada a través de las notificaciones de cambio hechas desde el modelo o como resultado de las peticiones del usuario.

El controlador es el encargado de recibir las peticiones del usuario e interpretarlas para realizar la acción adecuada y seleccionar la vista con la que se va a presentar la información.

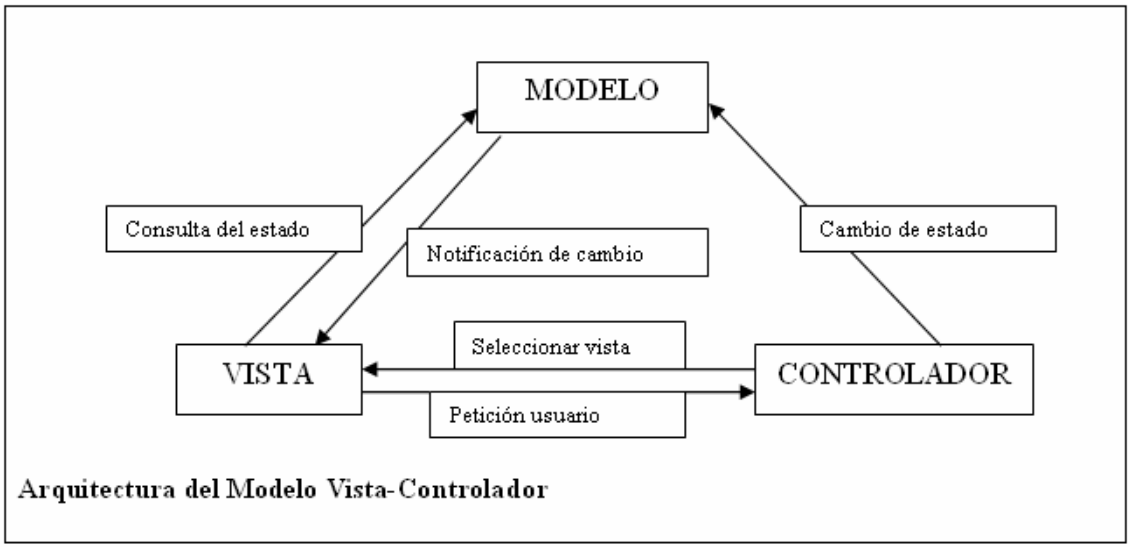

Figura 4: Arquitectura del patrón Modelo-Vista-Controlador

En el entorno utilizado, la Web, el modelo sólo puede responder a peticiones del controlador para generar una vista, pero no puede por sí solo notificar la actualización de la vista cuando se produce una modificación en los datos en el modelo.

3.1 Chasqui

Con estos conceptos de *Objeto de Aprendizaje* (digitalización objetos reales) surge el museo de Arqueología de la Facultad de Geografía e Historia, bautizado como Chasqui.[10]. Chasqui es fruto de la cooperación entre dos Facultades de la Universidad Complutense de Madrid: la Facultad de Geografía e Historia, con el departamento de Historia de América II (Antropología de América) y la Facultad de Informática con miembros del Grupo de Ingeniería del Software e Inteligencia Artificial del departamento de Sistemas Informáticos y Programación.

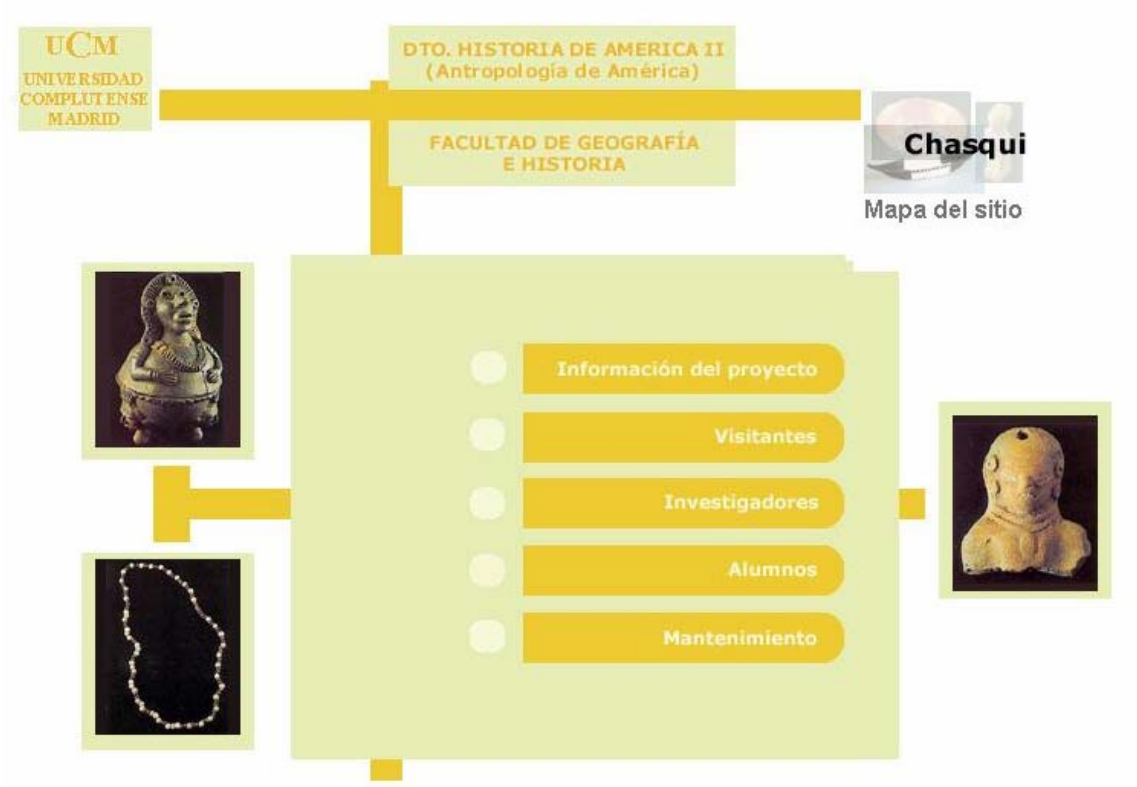

Figura 5: Museo Virtual Chasqui

Se empieza a trabajar en Chasqui en 2002 y es el primer antecedente del MIGS 1.0. Este museo arqueológico, sirve de apoyo a la docencia, al trabajo de los investigadores y a la creación de puntos de información para difusión cultural, cuya finalidad última es convertir en recursos educativos digitales los materiales arqueológicos y etnográficos del museo de Arqueología.

Esta herramienta de gestión es utilizada como una herramienta de trabajo por investigadores y alumnos como apoyo en la docencia de varias asignaturas y doctorados, lo que implica que la información que almacena su base de datos crezca rápidamente. Esto dificulta la gestión de los objetos que lo conforman, así como el mantenimiento de la propia aplicación. Entre los objetos de este museo hay una gran cantidad de documentos y recursos de aprendizaje, pero no pueden ser exportados a otros sistemas.

3.2 MIGS v 0.0

En septiembre del 2004, se empieza a trabajar en la versión 0.0 del Museo virtual de Informática, que recibe el nombre de MIGS, en honor a José García Santesmases catedrático de la Facultad de Informática.

Este museo, al igual que Chasqui, pretende digitalizar el museo físico de la facultad y convertirse en una herramienta de ayuda en la enseñanza e investigación informática.

Figura 6: Prototipo MIGS 0.0

MIGS 0.0 es el padre del actual MIGS, e intenta facilitar la gestión y mantenimiento del museo, mejorando el diseño de su base de datos para que toda la información y el material educativo sea consistente.

Otra importante mejora es permitir que el contenido del museo sea accesible y exportable a otros entornos, gracias al empaquetado de los objetos basado en el modelo de agregación de contenidos propuesto por IMS-CP [2].

En junio del 2004, finalizan los trabajos del MIGS 0.0 y en septiembre de ese año, se retoma para adaptar los nuevos requisitos, analizar y diseñar nuevas funcionalidades y para su paso a producción. El principal objetivo de dicho proyecto será obtener una nueva versión del museo de informática denominado MIGS 1.0.

4 Desarrollo del Proyecto

Durante todos estos meses hemos estructurado el trabajo a realizar en distintas etapas de desarrollo en función de los objetivos a cubrir. Dichas etapas han sido Iniciación, Reingeniería e Ingeniería, y han sido llevadas a cabo tal como se especifica a continuación.

4.1 Etapa de Iniciación

Inicialmente nos propusimos la lectura y comprensión de la memoria del proyecto del año anterior "*Uso y Gestión de Objetos de Aprendizaje en la Web*" [9] para tener una primera visión del trabajo ya realizado por nuestras compañeras. Durante dicha etapa no accedimos al código, pues el principal objetivos fue la comprensión de los conceptos generales del *e-learning*.

Una vez obtuvimos una base de los conceptos necesarios para el desarrollo procedimos a volver la memoria centrándonos en los apartados del diseño y comparándolo con el código concreto con el fin de tener una visión más profunda de la aplicación. Durante dicha etapa profundizamos en nuestros conocimientos de aplicaciones Web y aprendimos el desarrollo de aplicaciones con PHP.

Paralelamente a estas tareas procedimos montar la aplicación en nuestros ordenadores personales para tener una versión con la que trabajar en local y así poder realizar diversas pruebas sobre el código pudiendo comprobar los resultados de las mismas.

Esta última tarea nos trajo diversos problemas debidos al problema de versiones entre el servidor de aplicaciones Apache, PHP y la base de datos MySQL y la dificultad añadida de la configuración entre el Apache y PHP para que funcionen correctamente con los parámetros que son necesarios para que MIGS funcione correctamente. Finalmente, y tras muchas pruebas fallidas conseguimos integrar los 3 entornos anteriormente citados utilizando las siguientes versiones:

- Servidor Web Apache 1.3.31
- PHP 4.3.3
- Base de Datos MySQL 4.0.17

Debido a la complejidad de esta instalación se ha incluido un apartado en la memoria que explica detalladamente los pasos a seguir para la misma.

4.2 Etapa de Corrección: Reingeniería

En este punto vamos a explicar como abordamos cada uno de los fallos que se detectaron en la aplicación tras la realización de múltiples pruebas realizadas a la misma. Haremos una distinción siguiendo las capas del patrón Modelo-Vista-Controlador que se sigue en el desarrollo de la aplicación, explicado en Antecedentes de MIGS.

4.2.1 Reingeniería de Capa de Modelo

- **Búsquedas**: La funcionalidad buscar no se ejecutaba correctamente. La búsqueda daba fallos o directamente no respondía a las especificaciones realizadas por el usuario en la plantilla de búsqueda. Los ficheros que tuvimos que modificar para la corrección de la funcionalidad se correspondían a la *capa de modelo*. Se encuentran dentro del directorio Buscar y son: Ficha.php y ResultadoBusqueda.php. El problema consistía básicamente en que no se realizaba una correcta distinción entre los atributos numéricos y los de texto en la plantilla que rellenaba el usuario para realizar la búsqueda. Esto causaba que la consulta que se hacía a la base de datos fuera errónea. por lo que los resultados no eran satisfactorios.

- **Atributos repetidos**: En la creación de *Objetos de Aprendizaje* hay que rellenar las distintas fichas que lo componen. Se comprobó que al introducir un atributo con el mismo nombre en dos fichas diferentes (Campos Generales, Características Físicas, Características Tecnológicas) del mismo *Objeto de Aprendizaje* se producían inconsistencias en la base de datos o se producían errores en la creación del objeto. Profundizando en el error producido averiguamos que si se trataba de un atributo de texto se mostraba un mensaje de error en la pantalla, mientras que si se trataba de un atributo numérico no se mostraba ningún error pero se almacenaba en la base de datos como un atributo de texto, error que se iba arrastrando desde ese momento. Además, comprobamos que en una misma ficha de atributos (Campos Generales, Características Físicas, Características Tecnológicas) se podían añadir 2 atributos con el mismo nombre, algo que también ocasionaba errores e inconsistencias en la base de datos. Se almacenaban como dos atributos distintos pero el valor almacenado era común para ambos. Para solucionar esta cuestión se tomó la decisión de aumentar la clave primaria de las tablas “atributos_texto” y “atributos_numéricos” de forma que el campo “posición” perteneciera también a la misma. De esta forma se permite que haya atributos con el mismo nombre en 2 fichas diferentes ya que no ocasionaría ningún error en la base de datos como antes. Sin embargo, de esta forma se sigue sin permitir los atributos repetidos dentro de una misma ficha, por lo que decidimos controlarlo en el propio código en la correspondiente *capa de modelo*, en caso de que se intenten introducir 2 atributos con el mismo nombre en la misma ficha (Campos Generales, Características Físicas ó Características Tecnológicas) se muestra un mensaje de error y el atributo no se añade a la pantalla de creación de la ficha en cuestión. A continuación mostramos cual es la estructura de las tablas de atributos tras la modificación realizada:

```
CREATE TABLE atributos_numericos (
  idov int NOT NULL,
  nom_atrib varchar(255) NOT NULL,
  valor varchar(50),
  unidades varchar(20),
  categoria varchar(50),
  posicion int NOT NULL,
```

```
  PRIMARY KEY  (idov,nom_atrib,posicion),
  KEY ind_atributos_numericos (idov),
  FOREIGN KEY (`idov`) REFERENCES `objeto_virtual` (`idov`)
) TYPE=InnoDB;

-- Table structure for table `atributos_texto`
--

CREATE TABLE atributos_texto (
  idov int NOT NULL,
  nom_atrib varchar(50) NOT NULL,
  valor varchar(50),
  categoria varchar(50),
  posicion int NOT NULL,
  PRIMARY KEY  (idov,nom_atrib,posicion),
  KEY ind_atributos_texto (idov),
  FOREIGN KEY (`idov`) REFERENCES `objeto_virtual` (`idov`)
) TYPE=InnoDB;
```

Como se puede observar, el campo posición, que es el que se refiere al número de ficha, aparece incluido en la clave primaria de cada una de las tablas consiguiéndose gracias a ello el efecto explicado anteriormente. Los ficheros modificados se encuentran en el directorio privado y son para cada uno de los tipos de objetos que existen (piezas, fotos y documentos) los que representan la 3ª pantalla que aparece en el proceso de creación o modificación de objetos de las fichas de datos (Ej:, para objetos de tipo pieza: f1piezascajas.php, f2piezascajas.php,... tanto del directorio formularioOV, como del directorio modificarOV) Para ello se utiliza una función a la que llaman todos estos ficheros que comprueba la repetición de los atributos en cuestión y que se encuentra también en el directorio Privado. La función citada se llama compRepetidos.php.

4.2.2 Reingeniería de Capa de Control

- **Cancelar "Ir a un Objeto"**: La funcionalidad que permitía ir a un objeto especificando su *identificador de objeto virtual* tenía un error en su acción *cancelar*, pues en lugar de devolver el control a la aplicación siempre actuaba como la acción *aceptar*. Para solucionarlo modificamos el código JavaScript que tenía asociada la *capa de control* del botón "ir a un objeto" de forma que al cancelar devuelva el control a la aplicación en todos los casos sin tener en cuenta el valor introducido en el cuadro de texto de la ventana. Este código JavaScript se encuentra en cada uno de los ficheros "botoneraIzquierda" existentes para cada uno de los diferentes tipos de usuarios.
- **Navegación por posición**: Tras investigar el código y hacer diversas pruebas nos cercioramos de que la navegación se hacía por posición de los *Objetos de Aprendizaje* en la base de datos, es decir, cada *Objeto Virtual* era referenciado por la posición que ocupaba en la tabla "objeto_virtual". Sin embargo esto era incorrecto. Pongámonos en la siguiente situación: un usuario Visitante (V) se encuentra visitando el museo, concretamente el objeto 50. Si concurrentemente un usuario Investigador (I) accede a la aplicación y elimina todos los objetos del 51 en adelante, cuando en usuario V pulse en Siguiente se producirá un

error. Esto afecta a las funcionalidades “Siguiente”, “Anterior” y “Actualizar”.Este problema se solucionó haciendo una navegación por identificador de objeto virtual en lugar de por posición. De esta forma, cuando pulsamos sobre los botones principio, anterior, siguiente, ultimo o actualizar lo que se envía por los formularios correspondientes es el identificador del objeto que se está viendo en ese instante, se calcula la nueva posición que ocupa ese mismo objeto y, en función de la acción que se quiera realizar, se calcula la posición que ocupa el objeto virtual que queremos visualizar. Una vez obtenida la posición correcta del objeto que queremos visualizar lo obtenemos sus datos de la base de datos y lo mostramos. Los ficheros que han tenido que sufrir modificación pertenecen a la *capa de control* y han sido control.php tanto de la carpeta privado como de la carpeta público, botoneraAbajo.php, plantillaFichasPrivado, plantillaFichasPublico y plantillaFichasRestringido.

- **Cambio de usuario**: Si al estar navegando en el museo como usuario investigador realizamos un cambio de usuario se nos permite el acceso al museo, pero internamente quedaba almacenado en sesión la información del usuario anterior. Esta incidencia daría problemas en un futuro, ya que, como se explicará a continuación, se tenía pensado incluir una gestión más avanzada de usuarios. Para solucionar dicho error tuvimos que modificar en la *capa de control* el fichero verificaLogin.php de la carpeta Login. El error se producía porque no se reseteaba la sesión del usuario anterior de forma que los parámetros que poseía la sesión seguían siendo los del anterior usuario. Ahora esto ya no ocurre porque mediante el comando ‘session_destroy()’ destruimos la sesión anterior y sus parámetros inmediatamente antes de crear una nueva con el comando ‘session_start()’.

4.2.3 Reingeniería de Capa de Vista

- **Nombre del primer recurso**: Al crear o modificar un objeto en el museo en la ficha de recursos el nombre del primer recurso siempre tenía por defecto el nombre de “Recurso 1” y no existía la posibilidad de cambiarlo, siendo evidentemente conveniente que este nombre pudiera ser elegido por el usuario. Esto fue solucionado haciendo que el nombre del primer recurso tuviera asociado un cuadro de texto en el que se podía escribir el valor que se deseara. Por defecto, aparece el valor de “Recurso1” por si al usuario le parece conveniente mantenerlo. Los ficheros modificados fueron los relacionados con las pantallas intermedia y finales de las fichas de recursos de cada uno de los distintos tipos de objetos (piezas, fotos y documentos) correspondientes a la *capa de vista*. (Ej:, para objetos de tipo pieza: f4Piezas.php y f4piezascajas.php tanto en el directorio formularioOV como en el directorio modificarOV).

- **Edición de Fichas**: Otro de los errores detectados en el museo fue la edición de los valores de las fichas. Durante la navegación, tanto en el usuario visitante como el investigador, los datos mostrados en las diferentes fichas de los objetos permitían situarse sobre ellos y sobrescribir su valor. Dicha incidencia era tan solo visual, puesto que los cambios en ningún caso se veían reflejados en la base de datos, pero aun

así se procedió a su corrección. Para solucionarlo inicialmente le añadimos un atributo a los cuadros de texto que indicaba que solo podían ser de lectura evitando así que se pudiera sobrescribir su contenido. Sin embargo, finalmente decidimos mostrar los datos de cada ficha en formato plano eliminado con ello los cuadros de texto, de esta forma, además de evitarnos el problema de la escritura en ellos hacemos que los datos se muestren en su totalidad con la ayuda de barras de desplazamiento en caso necesario. Anteriormente, con el uso de cuadros de texto, si su valor tenía una longitud superior al del cuadro de texto la única manera de ver el contenido en su totalidad era desplazarse con el ratón hacia la derecha lo que lo hacia claramente incómodo. Los ficheros que tuvimos que modificar para la corrección de estos errores pertenecían a la *capa de vista* y fueron dentro del directorio Fichas los ficheros Ficha.php y FichaRecursos.php.

4.3 Etapa de Mejoras: Ingeniería

En este punto vamos a explicar como desarrollamos las mejoras propuestas al comienzo del proyecto tras haber probado y observado la aplicación así como aquellas que surgieron durante el desarrollo del proyecto por parte del resto de colaboradores del *Museo Virtual*.

Para realizar estas mejoras hemos seguido la metodología del trabajo que se especifico en Antecedentes de MIGS. Todas las nuevas mejoras se han centrado en la necesidad de cambiar o extender la herramienta, es decir, estando en la etapa *virtualización* se identifica una mejora y nos dirigimos a la etapa de *operatibilidad* salvo en el caso de la mejora “perfiles de usuario”, en la cual se ha necesitado extender el modelo y por tanto la transición ha sido de la etapa de *virtualización* a la etapa de *análisis del dominio*.

Haremos una distinción en función de las capas modificadas según el patrón Modelo-Vista-Controlador.

4.3.1 Ingeniería de capa de modelo

- **Traducciones**: El estándar de LOM [5] utilizado para la especificación de los metadatos está en inglés, y en el prototipo inicial se conservó dicho idioma incluso para mostrarlo a un usuario visitante. Se propuso cambiar dichas etiquetas a la hora de mostrarlas en la navegación con el fin de que ésta fuera más amigable para el usuario. Ej.: el metadato *Classification* era mostrado al usuario como *Clasificación*. Para ello, se creó el fichero “filtro.php” en la *capa de modelo*. Su función “traducir” recoge el nombre del metadato en el idioma del estándar y lo traduce a castellano. El fichero en el que llamamos a esta función para mostrar las etiquetas de los metadatos traducidas es ‘pieFichas.php’.
- **Imagen no visible**: Algunos recursos del tipo imagen no tenían asociada la vista preliminar que les correspondía tal y como aparece en la siguiente captura de pantalla que nos proporcionaron: Debido a esto en la vista preliminar del objeto al considerar que no tenía ninguna imagen como recurso propio mostraba también una imagen por defecto.

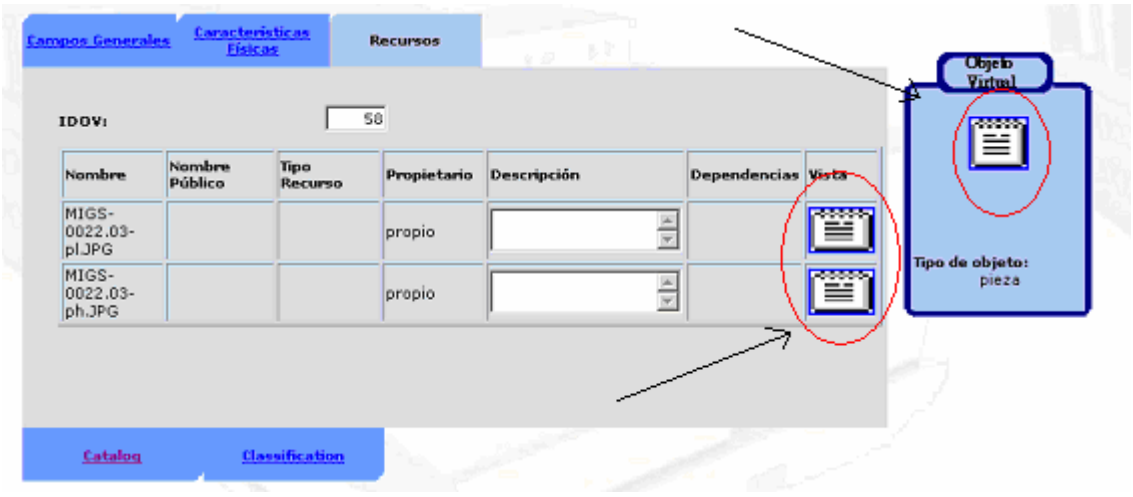

Figura 7: Imagenes no mostradas en MIG 0.0

Esto se debía a que la forma de obtener el tipo de los recursos no era correcta, ya que se consideraba que era el fragmento de la cadena del nombre que se encontraba a partir del primer punto. Por tanto, en los recursos que poseían varios puntos en su nombre el tipo se deducía que era el fragmento de texto que había desde el primer punto hasta el final de la cadena. Debido a esto no se obtenía un tipo de los que se consideran imágenes y por tanto el recurso era tratado como no imagen, lo que implicaba que las vistas mostraran una imagen por defecto y que la vista preliminar del objeto fuera otra imagen por defecto, al darse el caso de que no existía ninguna imagen como recurso propio del objeto. Para solucionarlo se buscó en la *capa de modelo* todas los lugares en los que se obtenía el tipo de los recursos y se adaptó el código para que obtuviera el tipo de los recursos teniendo en cuenta únicamente el fragmento de texto existente entre el último punto y el final de la cadena. El código correcto para obtener el tipo es el siguiente:

```
$divide = explode(".",$nombreRecurso);
$tipoRecurso = $divide[count($divide) - 1];
```

- **Nombre y tipo de recursos**: El nombre y el tipo de algunos recursos no aparecían rellenos a pesar de que si se habían introducido a la hora de crear el recurso en cuestión. Los colaboradores para solucionarlo tenían que modificar el objeto y volverlo a rellenar. Esta cuestión también se puede observar en la *Figura 7*. Esto también era debido a lo explicado anteriormente, ya que al calcularse mal el tipo se ejecutaba mal una consulta 'update' sobre la tabla recursos por lo que la fila del recurso en cuestión no se actualizaba. Una vez se solucionó el problema anterior los nuevos objetos introducidos ya se creaban correctamente.

4.3.2 Ingeniería de capa de control

- **Perfiles de usuario**: Se planteó hacer una gestión más avanzada de usuarios, en la que se contemplarían 3 tipos de usuario:
    - *Usuario Visitante*: Equivalente al usuario visitante de MIGS 0.0. Tan solo tendría permisos de navegación por el museo y los atributos que se le muestran están restringidos.

- *Usuario Administrador*: Equivalente al usuario investigador de MIGS 0.0. Además de las acciones del usuario visitante puede crear, modificar, exportar e importar cualquier objeto del museo. No se le restringe ningún atributo de los objetos.
- *Usuario Restringido*: Equivalente al usuario administrador con la salvedad que tan solo puede modificar aquellos objetos virtuales para los cuales tiene permiso. Todo objeto que cree en el museo se le asignarán automáticamente los permisos necesarios.

Para ello, fue necesario realizar cambios en el diseño de la base de datos. Se creo un nuevo campo en la tabla "usuarios" denominado "tipo", que indica el tipo de usuario. Este campo únicamente aparecerá relleno cuando el usuario en el que nos encontremos tenga permisos de administrador, teniendo como valor en ese caso `'administrador'`, en caso contrario se ignorará el valor del mismo.

También se tuvo que añadir una nueva tabla llamada "permisos", que almacena pares nombre de usuario-identificador de ov. El objetivo de la tabla es indicar que usuarios tienen permisos de edición sobre que objetos virtuales. Por defecto, cuando un usuario investigador sube un nuevo objeto al museo, se añade una tupla en esta tabla indicando que dicho usuario tiene permisos sobre el objeto recién subido. El administrador, tiene total libertad para añadir ó quitar permisos de edición a los usuarios,

Cuando un objeto virtual es borrado por un investigador o administrador por medio de la aplicación del museo los permisos sobre ese objeto son automáticamente eliminados.

```
CREATE TABLE usuarios (
  id int(11) NOT NULL auto_increment,
  usuario varchar(15),
  clave varchar(10),
  tipo varchar(15),
  PRIMARY KEY  (id)
) TYPE=InnoDB;


CREATE TABLE permisos (
  usuario varchar(15) NOT NULL,
  idov int NOT NULL,
  PRIMARY KEY  (usuario,idov),
  KEY ind_permisos (usuario),
  KEY ind_permisos2 (idov),
  FOREIGN KEY (`usuario`) REFERENCES `usuarios`
(`usuario`),
  FOREIGN KEY(`idov`) REFERENCES `objeto_virtual` (`idov`)
)
```

Los cambios necesarios han sido sobre la *capa de control*, concretamente sobre el fichero "control.php" del directorio "privado". También ha sido necesario crear dentro del directorio "Fichas" los ficheros 'PlantillaFichasRestringido' y el fichero 'botoneraIzquiedaRestringido' correspondientes a la *capa de vista*. En el fichero 'control.php' se comprueba si el usuario investigador actual tiene permiso sobre el objeto que se va a mostrar a continuación. En caso afirmativo se muestra

'PlantillaFichasPrivado', que muestra 'botoneraIzquierdaPrivado' con todas las funcionalidades de edición sobre el objeto actual. En caso negativo se muestra 'PlantillaFichasRestringido', que contiene 'botoneraIzquierdaRestringido' que muestra todas las funcionalidades excepto las de edición del objeto virtual. Para realizar la comprobación de los permisos de edición sobre un objeto virtual concreto y el usuario actual, en la *capa de modelo* se utiliza la función "puedeModificar" del archivo "basedatos.php".

4.3.3 Ingeniería de capa de vista

- **Nueva interfaz de navegación**: El prototipo que nos fue proporcionado podía tener una apariencia más llamativa, en especial la parte que sería utilizada por los usuarios visitantes ya que es la más habitual. En primer lugar la página principal ha cambiado de formato, ahora se muestra una pantalla de fondo negro con un dibujo de la facultad y tres enlaces, uno dirige a una breve presentación del museo, y los otros a las navegaciones para un usuario visitante ó investigador. La pantalla de acceso para los investigadores ha cambiado también de formato siendo ahora más amigable. Además, se nos indicó que la cabecera del museo debía tener otro aspecto con una apariencia más amigable introduciendo nuevos colores, una cenefa dinámica representando una tarjeta perforada, un acceso a una página de ayuda básica y otro que redireccione a la página principal del museo. El botón de ayuda nos dirige a una ventana auxiliar en la que se nos muestra una breve introducción sobre el contexto del proyecto y una breve guía de uso de la aplicación para los usuarios visitantes.

  En cuanto a la navegación, los botones del menú lateral izquierdo han sido sustituidos por otros más atractivos y las fichas tienen una nueva apariencia con un fondo ajustado al nuevo estilo que le queríamos dar al museo y un formato de texto plano en lugar de cuadros de texto, mejorando la visualización de los datos.

  Otro cambio significativo es el de la ficha de recursos. Para los usuarios visitantes ha quedado restringida únicamente a la información del tipo, descripción y vista preliminar del recurso. Además, la apariencia es distinta que para las demás fichas puesto que se muestran los datos en formato de tabla y la pestaña correspondiente a la ficha es de un color más oscuro.

  Los botones de navegación situados en la parte inferior de la pantalla también han cambiado su aspecto para adaptarse al mismo formato que los del nuevo menú lateral.

  Por otra parte, la vista preliminar de cada objeto, situada en la parte derecha de la pantalla, tiene una nueva apariencia representada con una tabla de borde fino, un fondo diferente y nuevos datos añadidos, ya que ahora, además de mostrar los datos del primer recurso propio que sea imagen, se muestra también una vista preliminar del recurso y una breve descripción del objeto. En caso de no existir este recurso imagen se muestran únicamente los datos del objeto. Asimismo, en la ficha de recursos, tras pinchar sobre la vista preliminar de cada uno de ellos, también se muestran los datos del objeto al que pertenece dicho recurso con un formato idéntico al explicado para la vista preliminar del objeto.

Para permitir una mejor identificación del objeto visitado actualmente, el número de ficha de la base de datos en la que nos encontramos la hemos pasado a situar debajo de la vista preliminar del objeto en lugar de estar debajo de la botonera de la izquierda.

La funcionalidad buscar también ha cambiado de apariencia. En primer lugar hemos cambiado el color de fondo de la plantilla de búsqueda y hemos colocado tres pares de botones situados en la parte superior, central e inferior de la misma para que el usuario pueda acceder a las funcionalidades desde cualquier parte de la pantalla haciéndolo así más cómodo. Una vez que la búsqueda se ejecuta se muestra una pantalla en la que aparecen los resultados. En ella hemos cambiado de apariencia el botón buscar y hemos mostrado el mensaje que indica el número de resultados obtenidos en la parte superior de la pantalla en lugar de en la parte inferior.

Tanto las pantallas de la búsqueda como las pantallas de navegación y entrada para usuario investigador poseen un fondo diferente al que había anteriormente. En el se muestra una imagen de contorno de la Facultad de Informática de la Universidad Complutense de Madrid

Es necesario resaltar que las imágenes que hemos colocado en la página principal, las de los botones de navegación y todas aquellas que hemos usado han sido realizadas por Sara Olmos, diseñadora gráfica profesional. Nosotros las hemos retocado, adaptado a la apariencia del museo e integrado al código de la aplicación. Asimismo, cabe destacar que todos los estilos usados, han sido especificados y aplicados a partir de una hoja de estilos, de forma que si en un futuro se desean modificar los formatos éstos puedan ser retocados fácilmente.

Todos estos cambios han tenido lugar en la *capa de vista*, y se pueden observar los resultados en el Manual de Usuario.

- **Vista preliminar**: Un aspecto de la aplicación a mejorar era la vista preliminar de los objetos que aparece en la parte derecha de la pantalla así como la de los recursos del mismo. En cuanto a los recursos se consideró más apropiado que al pinchar sobre el enlace que poseen en su vista se mostrara además de los datos del recurso una descripción del objeto al que pertenecen. Todos los cambios pertenecen a la *capa de vista*, más concretamente, si el recurso en cuestión es una imagen se llama al fichero 'MuestraImagen.php' que presenta una ventana dividida en 3 celdas, 2 en la parte superior y 1 en la parte inferior. En la parte superior izquierda aparece una vista de la imagen, que se puede redimensionar, en la parte superior derecha aparecen los datos de la imagen (Nombre, tipo, propietario y descripción del recurso) mientras que en la parte inferior se muestran algunos datos del objeto propietario (Nombre, Categoría, Ubicación y Descripción). Si el recurso no es una imagen se llama al fichero "MuestraRecurso.php", que muestra los datos del recurso en cuestión y del objeto propietario de forma análoga al anterior pero además abre una nueva ventana en la que abre el recurso. En el caso de que el recurso no sea una imagen se muestra en el enlace de su vista una imagen por defecto mientras que en caso contrario se muestra la imagen en miniatura.

  En lo que respecta a la vista preliminar del objeto se procede de la

siguiente forma: en caso de que el objeto posea un recurso propio que sea imagen se muestra el fichero "MuestraRecurso.php" mientras que si esto no ocurre se llama al fichero "MuestraResumen.php" que muestra únicamente los datos del objeto (la descripción del objeto y sus atributos numéricos y de texto en caso de que existan.). En el caso de que el objeto si disponga de un recurso propio que sea una imagen se muestra en el enlace la imagen en miniatura mientras que si se da el caso contrario se muestra una imagen por defecto.

Por tanto los ficheros modificados con respecto a esta funcionalidad han sido "MuestraRecurso.php" y "MuestaImagen.php". Los resultados pueden observarse en el Manual de Usuario.

- **Menús dinámicos**: En el momento de la creación de un *Objeto Virtual* algunos de los campos suelen tener siempre los mismos valores, por ello nos sugerían que sería más conveniente un menú desplegable en lugar de un cuadro de texto para introducir el valor del campo. De esta manera se producirían menos errores debido a que los valores ya estarían escritos correctamente en el menú desplegable de forma que tan sólo habría que seleccionarlos. Además, nos proponían que el menú debería ser configurable, de manera que se pudieran introducir nuevos elementos al mismo en función de las necesidades que fueran surgiendo. Debido a las grandísimas ventajas que esta mejora podía traer a la hora de la creación y modificación de objetos se decidió llevarla a cabo. Para ello recibimos una lista por parte de los colaboradores de los atributos y metadatos para los cuales querían menús desplegables así como los valores que querían que hubiera por defecto para cada uno de ellos. Para ello tuvimos que modificar los archivos referentes a cada uno de los tipos de objetos (piezas, fotos y documentos) en la 3ª pantalla de creación y modificación de objetos afectanto sobre todo a la *capa de vista* aunque también ha sido modificada parcialmente la *capa de modelo*. En los casos en los que nos pidieron que creáramos un menú desplegable para atributos que no existían ya en las fichas de creación y modificación los cambios que hubo que realizar fueron más profundos teniendo que modificar también la 1ª y 2ª pantalla de creación y modificación para añadir en los formularios los nuevos atributos. Este caso se nos dio por ejemplo en la categoría de piezas, que no existía,. teniendo que modificar tanto en modificarOV como en formularioOV los ficheros "Piezas.php", "f1piezas.php" y "f1piezascajas.php" En estos menús se muestran siempre los valores que nos han fijado los colaboradores y además los diferentes valores que posee dicho atributo en la base de datos para esa categoría de objetos (Si nos encontramos creando o modificando un objeto de tipo pieza se mostraran los valores de ese atributo que haya en la base de datos que pertenezcan a objetos de tipo pieza y que por tanto han sido elegidos en otra ocasión anterior). Además del menú desplegable se creó un cuadro de texto a su derecha con el nombre "Otra" para introducir, en caso de que el investigador no desee elegir un valor de los ya presentes en el menú, otra entrada al menú. Si añade una entrada nueva ésta queda seleccionada por defecto en ese momento, aunque el investigador puede elegir otra en todo momento. Pulsando sobre aceptar y finalizar, en ese orden, el valor seleccionado se propaga a la pantalla intermedia. En caso de que se quiera volver a editar la ficha el

valor que aparece seleccionado por defecto es el que se había seleccionado anteriormente. Un hecho a dejar claro es el de que un elemento añadido a un menú desplegable no aparecerá en el menú en la creación o modificación de otro objeto si finalmente el objeto para el que se añadió no se creó con ese valor para ese atributo.

Cabe destacar que hemos creado un menú desplegable en la ficha de metadatos 'Ciclo de vida' que no posee un cuadro de texto para añadir nuevos valores en el menú. Esto es debido a que se nos indicó que ese metadato únicamente podía tener uno de esos valores prefijados y ningún otro. Este metadato en cuestión se llama 'Evento' y se ha incluido de esta manera en los 3 tipos de objetos (Piezas, Fotos y Documentos). Otro aspecto a destacar es que en el menú desplegable 'tipo' presente en la ficha de recursos se muestran, además de los valores por defecto indicados por los colaboradores, los diferentes valores existentes en la tabla de recursos para su campo tipo pero sin distinguir el tipo del objeto en el que nos encontramos.

4.4 Nuevo diseño de la base de datos

El Modelo Entidad/Relación de la base de datos que se ajusta a lo explicado anteriormente es el siguiente:

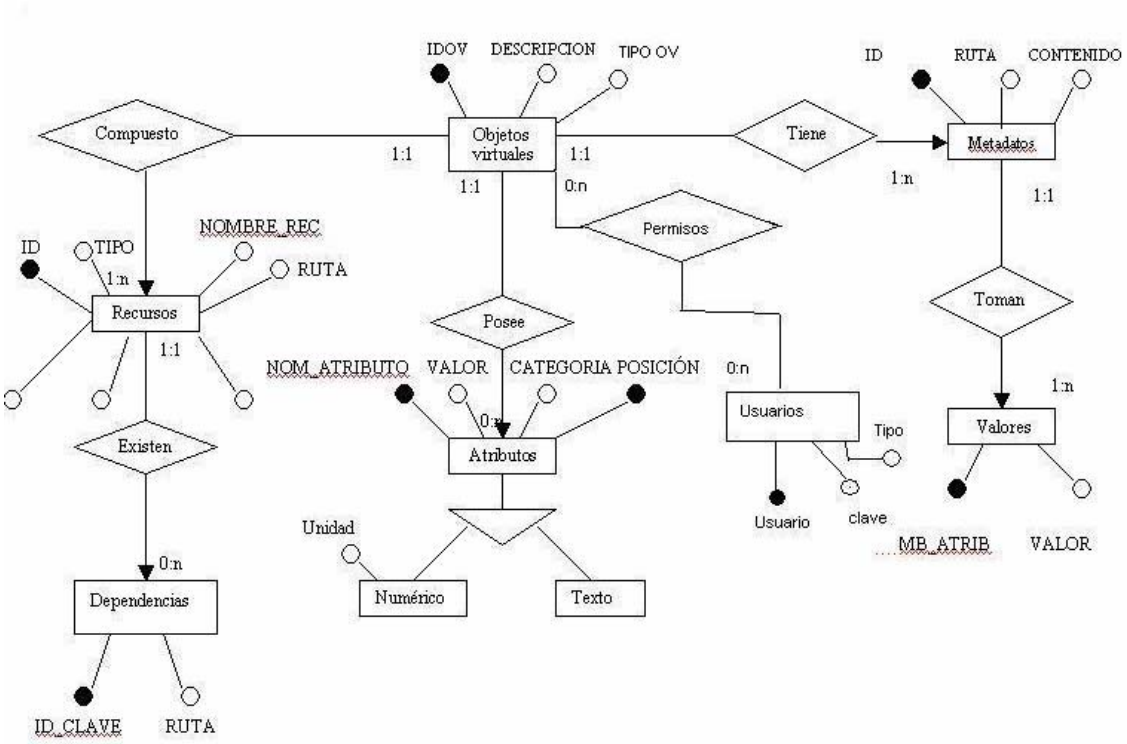

Figura 8: Diagrama Entidad/Relación MIGS 1.0.

El Modelo Relacional que se corresponde con el diagrama de la Figura 8 es el siguiente:

1. Objeto virtual:
    - idov: entero.
    - descripcion: texto.
    - tipoOV: String de tamaño 10.

- Clave primaria: idov.

2. Metadatos
    - id: entero.
    - ruta: String de tamaño 255.
    - Contenido: String de tamaño 255.
    - num_ruta: String de tamaño 20.
    - Clave primaria: id.
    - Clave externa: idov perteneciente a la tabla objeto_virtual.
3. Atributos_metadatos
    - id: entero.
    - nom_atrib: String de tamaño 50.
    - valor: String de tamaño 30.
    - Clave primaria: id, nom_atrib.
    - Clave externa: id perteneciente a la tabla metadatos.
4. Atributos_numericos
    - idov: int.
    - nom_atrib: String de tamaño 255.
    - Valor: String de tamaño 50.
    - unidades: String de tamaño 20.
    - Categoría: String de tamaño 50.
    - posicion: entero.
    - Clave primaria: idov, nom_atrib,posición.
    - Clave externa: idov perteneciente a la tabla objeto_virtual.
5. Atributos_texto
    - idov: entero.
    - nom_atrib: String de tamaño 50.
    - valor: String de tamaño 50.
    - categoria: String de tamaño 50.
    - posicion: entero.
    - Clave primaria: idov, nom_atrib, posición.
    - Clave externa: idov perteneciente a la tabla objeto_virtual
6. Recursos
    - id: entero.
    - idov: entero.

- nom_rec: String de tamaño 60.
- nom_rec_publico: String de tamaño 60.
- tipoRec: String de tamaño 60.
- descripcion: texto.
- tipo: String de tamaño 50.
- ruta: String de tamaño 50.
- Clave primaria: id.
- Clave externa: idov perteneciente a la tabla objeto_virtual.

7. Dependencias
    - id_clave: entero.
    - ruta: String de tamaño 100.
    - Clave primaria: id_clave.
    - Clave externa: id perteneciente a la tabla recursos.
8. Usuarios:
    - Usuario: String de tamaño 15.
    - clave: String de tamaño 10.
    - tipo: String de tamaño 15.
    - Clave primaria: usuario.
9. Permisos
    - Usuario: String de tamaño 15.
    - idov: entero.
    - Clave primaria: usuario, idov.
    - Clave externa: usuario perteneciente a la tabla usuarios.
    - Clave externa: idov perteneciente a la tabla objeto_virtual.

5 Estado Actual y Objetivos Futuros

Tras la finalización del código del acceso web al Museo de Informática García Santesmases y el visto bueno de la Facultad de Informática de la Universidad Complutense de Madrid, dicho organismo ha decidido poner en producción los resultados de este proyecto. Se puede tener acceso a él a través de la página oficial de la facultad http://www.fdi.ucm.es, siendo también posible el acceso directo en http://www.fdi.ucm.es/migs.

Esto no significa que dicho proyecto se de por terminado y no admita ampliaciones y mejoras. Por nuestra parte proponemos los siguientes objetivos para los futuros desarrolladores de nuevas versiones de MIGS:

- **Interfaz de Administrador**: Una de las nuevas funcionalidades incorporadas a MIGS 1.0 ha sido el control de permiso de edición de los distintos *objetos de aprendizaje* por parte de los usuarios investigadores, teniendo aparte usuarios con permisos de administración. Aunque al introducir nuevos *Objetos de Aprendizaje* en el museo las tablas son actualizadas automáticamente, para asignar o retirar permisos sobre los objetos es necesario realizarlos sobre las tablas de la base de datos. Por eso mismo proponemos una interfaz accesible exclusivamente por usuarios con permisos de administrador para una cómoda administración de los permisos del resto de usuarios, así como tener la posibilidad de crear nuevos usuarios o eliminarlos.

- **Registro de Accesos**: Para una depuración de los errores que pudieran surgir durante la puesta en producción de MIGS sería aconsejable almacenar los pasos que un usuario realiza en el museo, o al menos las acciones más relevantes, ya sea en la base de datos o en ficheros de texto plano. La administración de dichos registros podrían incorporarse a la interfaz de administrador para una mayor comodidad y filtración de los resultados.

- **Navegación Avanzada**: Como ya se ha expuesto, los *Objetos de Aprendizaje* de MIGS están clasificados en tres categorías: pieza, foto y documento. Proponemos un tipo de navegación que no este basado tan solo en el identificador del *Objeto Virtual*, sino que también sea posible realizarla filtrando algunos de los elementos por la categoría de estos o incluso por los metadatos de los mismos.

- **Vista Global**: En los recursos de los *Objetos de Aprendizaje* de MIGS cabe la posibilidad de estar compuestos a su vez por otros *Objetos de Aprendizaje* del museo. Actualmente existe una navegación a través de la ficha recursos, pero sería interesante poder visualizar en forma de árbol dichas dependencias.

- **Interfaz de investigador más cómoda**: Uno de los objetivos de este proyecto era mejorar la interfaz de MIGS 0.0, habiendo sido alcanzado en los apartados de navegación y búsqueda. En cuanto a la parte investigadora deja por desear una navegación más intuitiva y que esté ligada a los nuevos estilos que MIGS 1.0 proporciona en la parte de navegación.

- **Generador de museos**: MIGS es un museo virtual específico para la Facultad de Informática de la Universidad Complutense de Madrid, pero podría tener una posibilidad aun más ambiciosa: tratarse de un esqueleto para que cualquier organismo lo configure para su uso personalizado. Por eso mismo proponemos un museo que no esté estructurado de una manera estática, sino que permita amoldarse a las necesidades de otros clientes de una forma cómoda y rápida. Así pues se podría construir un generador de museos que escriba automáticamente los archivos de la aplicación, o aun mejor, especificar la estructura del museo de una forma dinámica, que pueda ser cambiada en cualquier momento permitiendo su importación / exportación en formato XML.

6 Manual de Usuario

Este apartado explica el funcionamiento del Museo Virtual de Informática García Santesmases, detallando todas las funcionalidades a las que cada usuario tiene acceso.

Una vez realizada la instalación del museo tal como se refleja en el manual de instalación es posible acceder a la página principal de MIGS abriendo un navegador con http://localhost/migs/principal/principal.html. También es posible acceder a la versión pública del museo desde la página oficial de la Facultad de Informática de la Universidad Complutense de Madrid http://www.fdi.ucm.es.

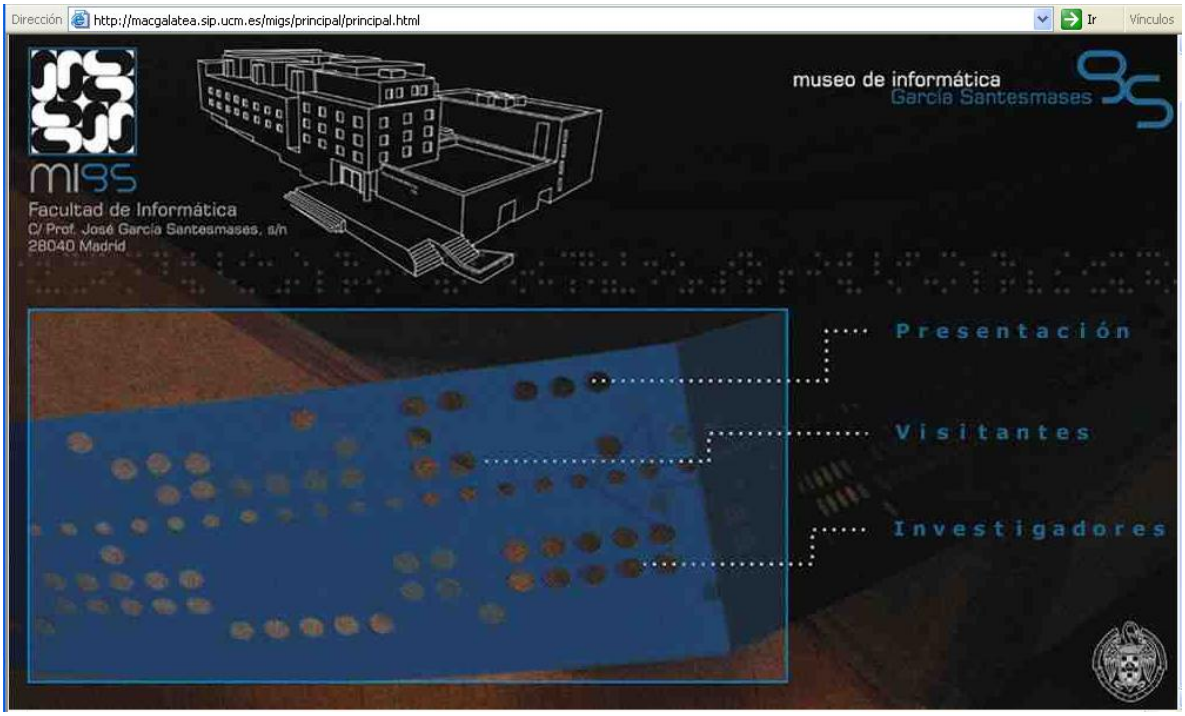

Figura 9: Página principal del Museo García Santesmases.

En esta página existen tres enlaces:

- **Enlace Presentación**: Se accede a una explicación del museo real de la facultad de informática con datos de interés:

Figura 10: Página de Presentación.

- **Visitantes**: Permite tanto la visualización de todos los objetos del museo como la realización de búsquedas sobre estos.
- **Investigadores**: Además de las funcionalidades del usuario visitante tienen acceso a las funcionalidades de creación, modificación y descarga de los objetos del museo. Si el usuario no es administrador tan solo podrá editar los objetos para los cuales no tiene el acceso restringido.

6.1 Usuario Visitante

Se trata de un usuario que puede ver los objetos del museo pero no tiene permisos para su manipulación. Su funcionalidad principal es la navegación por los distintos objetos del museo y la realización de búsquedas en base a las características de éstos.

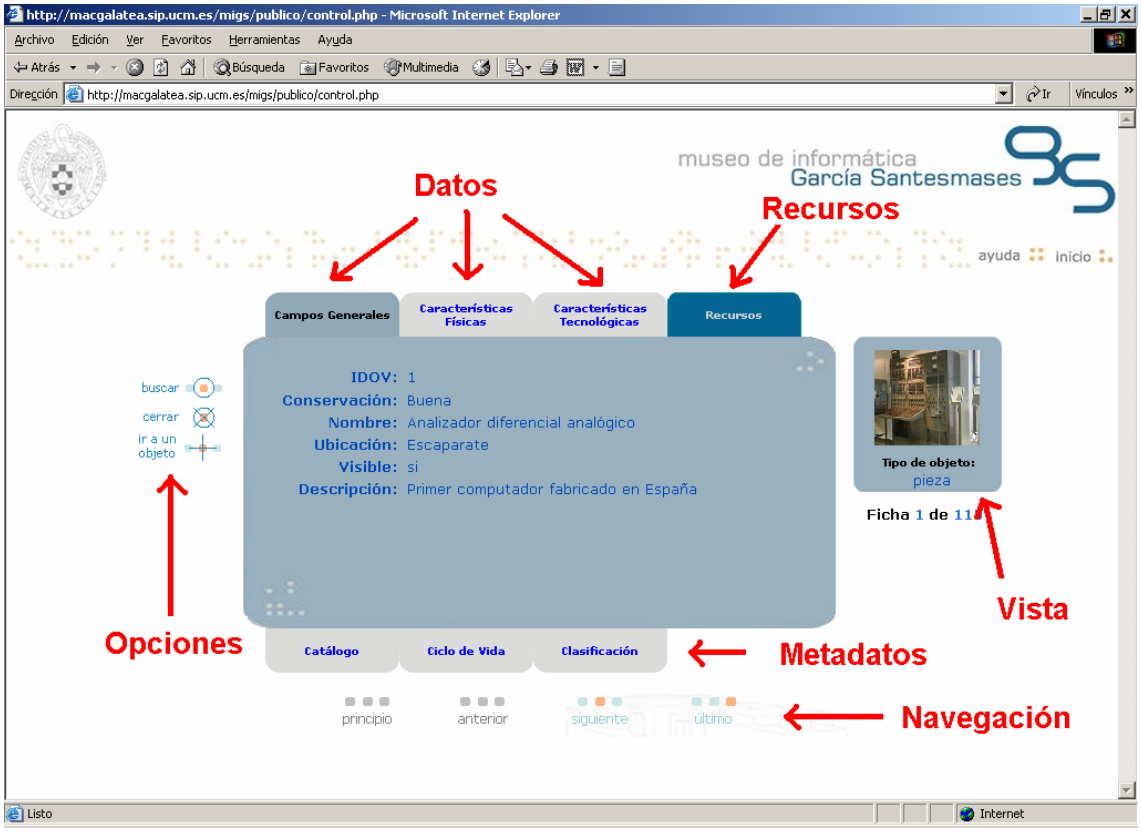

Figura 11: Vista de un objeto virtual para un usuario visitante.

El la parte superior de la Figura 11 se puede observar la cabecera del museo con enlaces a la ayuda del mismo y a la página principal del museo.

En la parte central de la Figura 11 se encuentra la ficha del objeto virtual que se está visualizando así como enlaces al resto de fichas que lo componen. En la parte superior están las fichas de datos y la de recursos, y en la inferior las de metadatos.

En el menú de la izquierda se encuentran las distintas acciones permitidas en función de los permisos de usuario actual, y a la derecha tenemos una vista preliminar del objeto.

En la parte inferior, están los botones de navegación que permiten desplazarse por el museo.

6.1.1 Navegación por el museo.

Las páginas de los objetos del museo consisten en un conjunto de fichas rellenas con los datos del objeto, tal como se puede observar en la Figura 11. En todos los objetos no se visualizan el mismo número de fichas, ya que sólo aparecen aquellas fichas que tienen sentido para ese objeto, es decir las fichas que contienen algún dato que mostrar.

Existen tres tipos de fichas:

- **Datos**: contienen datos característicos del propio objeto, en este tipo de fichas se encuentran “Campos generales”, “Características físicas” y “Características tecnológicas”. Están ubicadas en la parte superior de la ficha actual.

- **Recursos**: Constituyen el conjunto de elementos informativos asociados a un objeto. Esta ubicada en la parte superior de la ficha actual con un color más oscurecido.
- **Metadatos**: Describen el contenido, calidad, condiciones y otras características de los datos, permitiendo así a una persona ubicar y entender los datos. Las fichas que representan los metadatos son "Clasificación", "Catalogación" y "Ciclo de Vida", y se encuentran ubicadas en la parte inferior de la ficha actual.

Para acceder a cualquiera de las fichas se pulsa sobre la pestaña de cada una. En la Figura 11 se muestra la ficha **Campos Generales** del objeto primero de la base de datos, que será lo primero que vea un usuario que acceda al museo.

A la derecha del objeto, siempre aparece un recuadro con una imagen y debajo de ella tenemos información del tipo de objeto virtual que actualmente se está visualizando. La imagen que aparece, es la imagen del primer recurso propio del objeto actual que sea imagen. Pinchando en el objeto que muestra la Figura 11 aparece una nueva ventana con algunos datos del objeto y con datos del recurso al cual pertenece la fotografía.

Figura 12: Vista preliminar de un recurso propio con imagen.

Navegando por el museo podemos ver que algunos objetos no disponen de recursos propios que sean de tipo imagen. En estos casos el objeto se verá representado con una imagen representando un cuaderno tal y como se aprecia en la Figura 13.

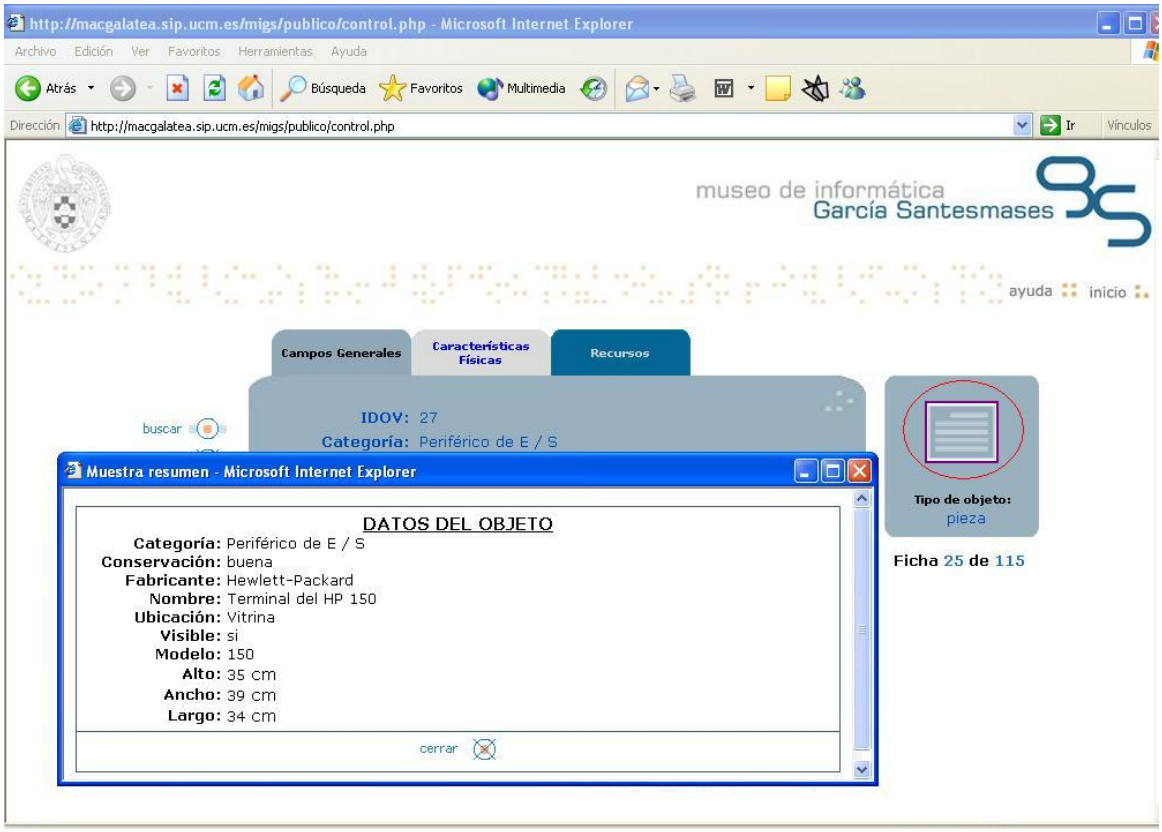

Figura 13: Vista preliminar de un objeto sin recursos propios imágenes.

Una de las fichas más interesantes es la ficha de recursos, la cual permite acceder a diferentes recursos y objetos del museo que están relacionados con el objeto actual. De esta forma se puede realizar una navegación entre distintos objetos que tienen algún tipo de vinculación. Vemos la ficha de recursos del idov 1, la cual tiene recursos de varios tipos:

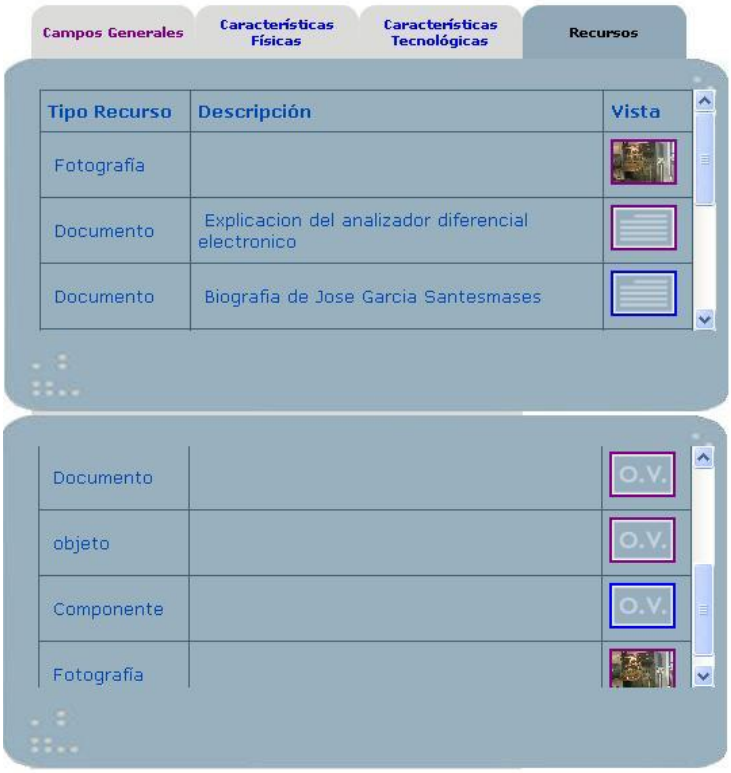

Figura 14: Vista de la ficha recursos de visitante.

Como estamos en el museo con un usuario visitante, esta ficha no muestra toda la información de cada recurso. Cuando veamos investigador, se mostrarán los demás campos (propietario, nombre privado…).

6.1.2 Ir A

Permite visualizar un objeto virtual determinado especificando su identificador. Si no existe un objeto con dicho identificador se muestra un mensaje de error avisando de la situación.

6.1.3 Buscar Objetos

Además de la navegación por el museo objeto por objeto, un usuario también tiene posibilidad de buscar objetos con unas características específicas. Pulsando sobre el primer botón de la botonera izquierda, “Buscar”, se llega a la página de la Figura 15:

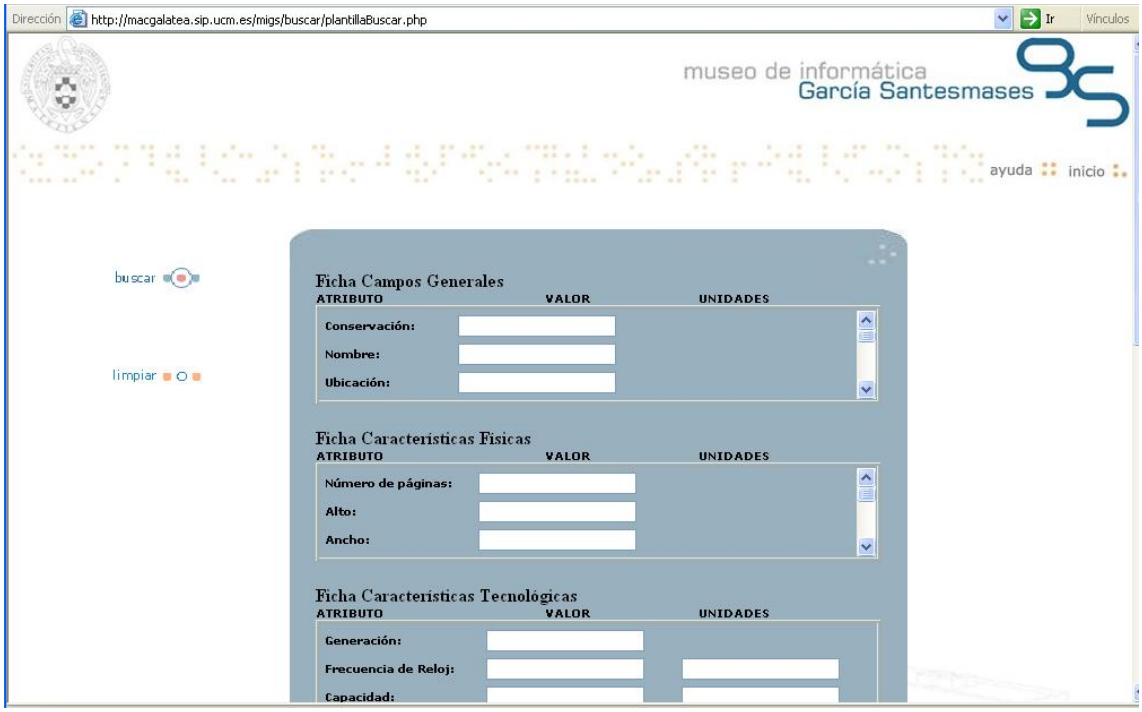

Figura 15: Buscar un objeto en el museo por atributos.

Cuando el administrador o un investigador crea un nuevo *Objeto Virtual* puede añadir atributos distintos a los que vienen por defecto. Esta ficha es dinámica, ya que se puede buscar por todos los atributos que tienen los objetos virtuales: los que aparecen por defecto y los nuevos que los investigadores pueden haber añadido.

Una vez que se han introducido todos los atributos necesarios para una búsqueda determinada, se puede pulsar sobre cualquier botón “buscar”.

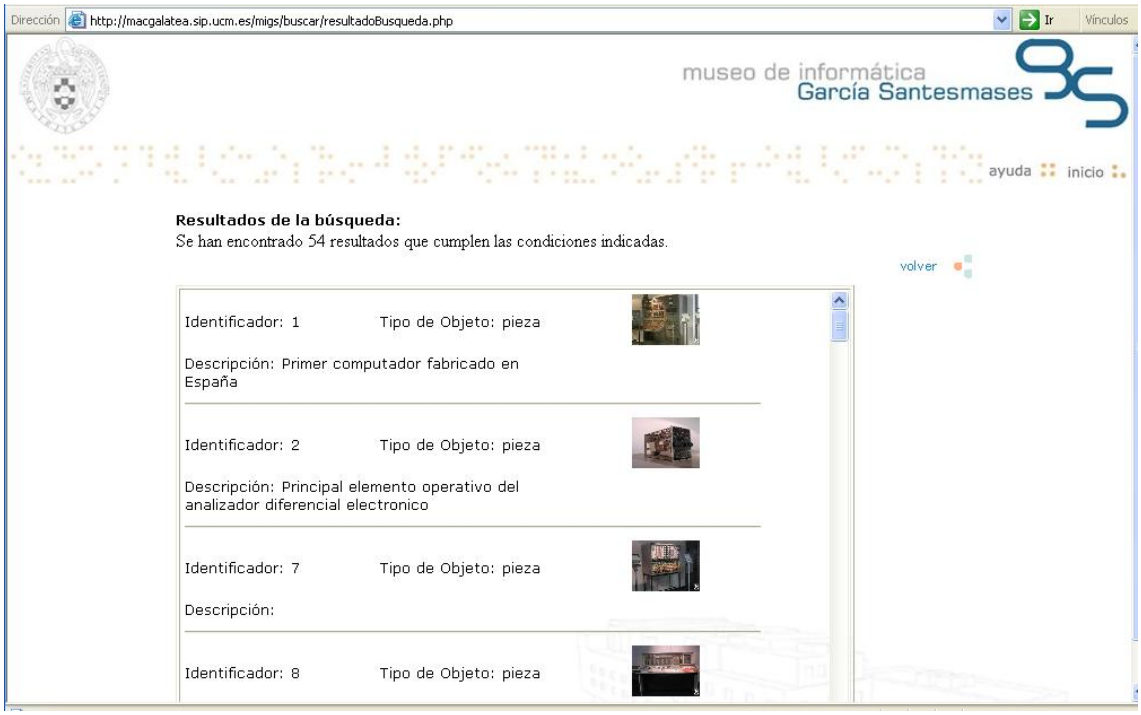

Figura 16: Página de resultados de una búsqueda.

6.2 Usuario Investigador

Para acceder al museo como usuario investigador, hace falta un identificador y contraseña. Hay un enlace a investigadores desde la página principal:

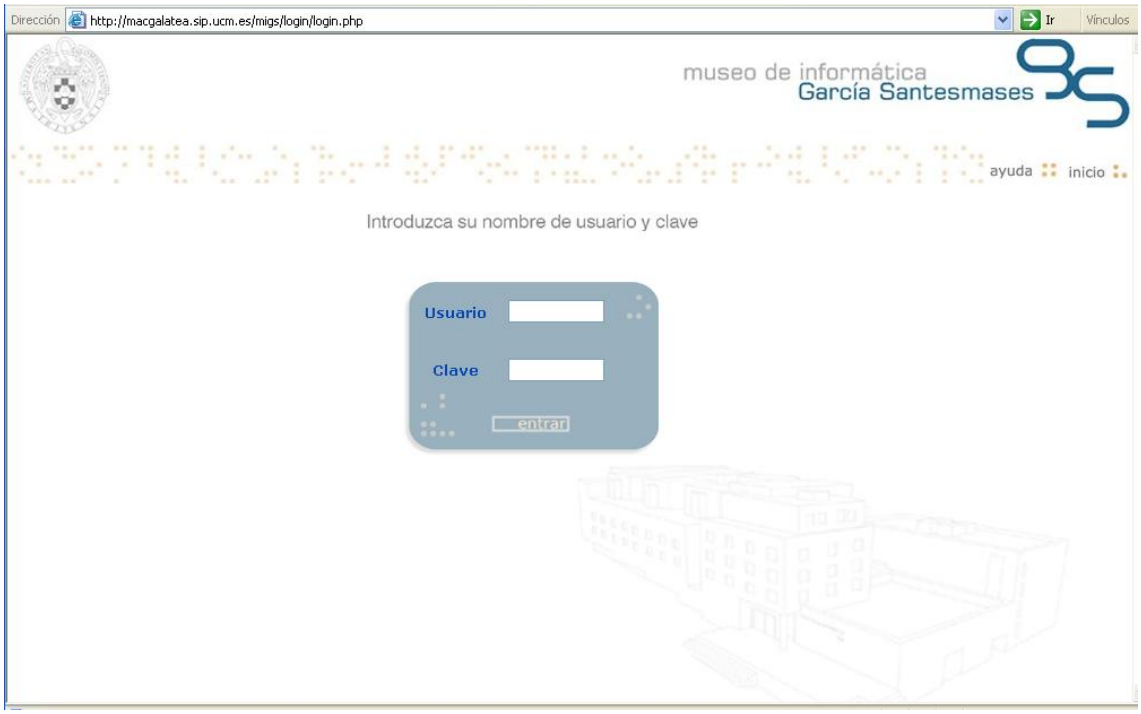

Figura 17. Acceso al museo como investigador

El usuario investigador tiene la capacidad de crear nuevos objetos virtuales, así como de subirlos y bajarlos del museo en formato zip. También puede modificar aquellos objetos virtuales sobre los que tenga permisos de edición.

Un investigador tiene permisos por defecto sobre los objetos que crea, pero el resto de objetos del museo no puede modificarlos ni eliminarlos a no ser que se le otorguen explícitamente los permisos necesarios. De esta forma, un investigador, al navegar por los objetos, según tenga o no permisos de edición sobre el objeto actualmente visitado, la apariencia del objeto será como la de visitante si no tiene privilegios sobre él o como la de la Figura 18, si dispone de permisos de edición.

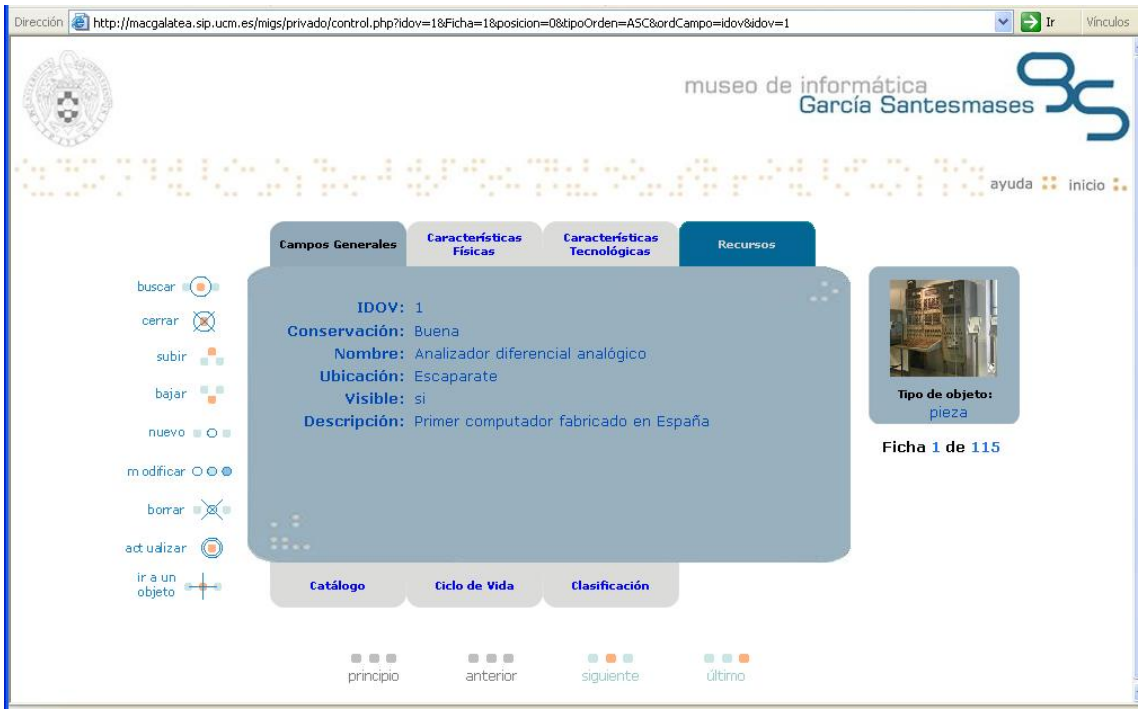

Figura 18: Vista de un objeto virtual para un usuario investigador.

Se observa que es la misma información que para el usuario visitante pero que aparecen más botones en el lado izquierdo.

El **administrador** es un tipo de usuario, el cual tiene permisos de edición sobre cualquier ov del museo. También este usuario administrador puede proporcionar o quitar privilegios de edición a investigadores sobre otros objetos.

En un trabajo futuro, se debería implementar una aplicación para que el administrador pueda gestionar los permisos de los investigadores sobre los objetos del museo, ya que actualmente, el administrador debe tocar directamente una tabla de la base de datos sobre la que se asienta todo el migs, con los riesgos que podría conllevar.

La información que aparece en las fichas, es la misma que aparece si se conecta un visitante. La única ficha en la que aparece más información sería la de recursos, como ya comentamos anteriormente. Los visitantes solo ven en esta ficha, el tipo de objeto, la descripción y la vista. Un investigador, además de esto, ve el nombre privado (el nombre del fichero del recurso), el nombre público, las dependencias de ese recurso con otros recursos ú objetos virtuales del museo así como el propietario del mismo.

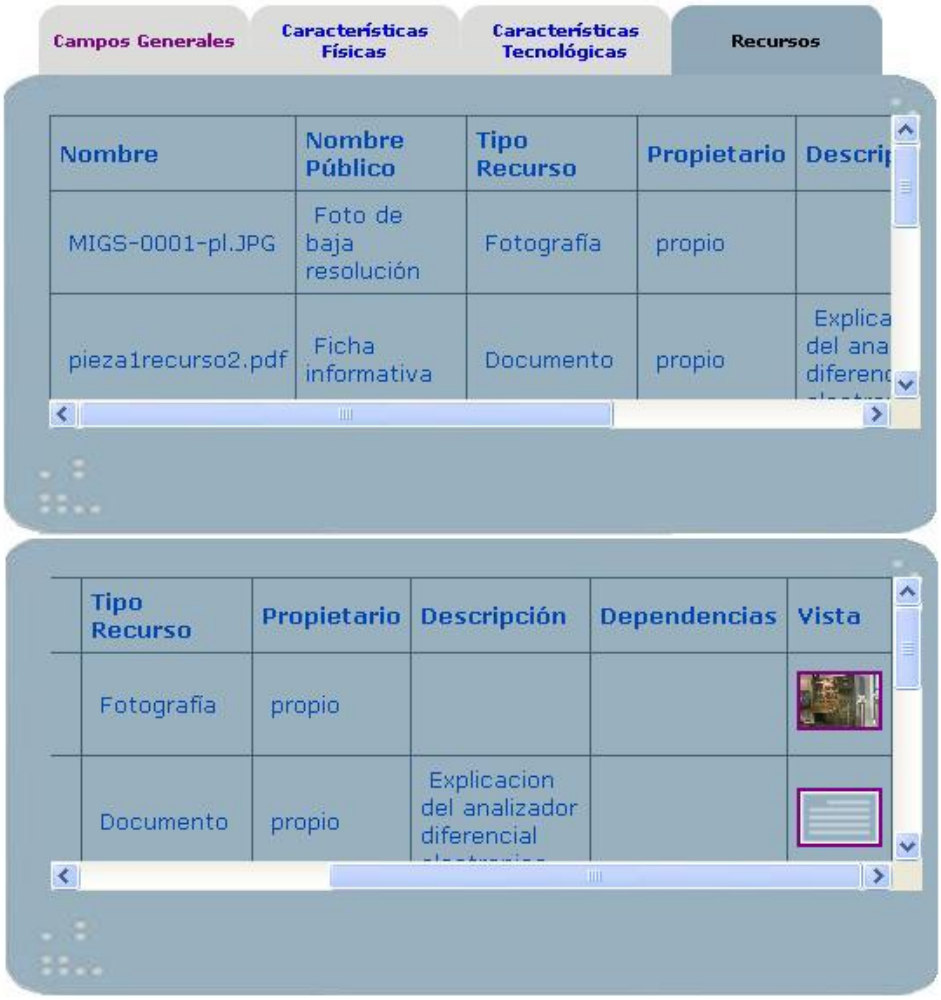

Figura 19: Vista de la ficha recursos de investigador.

6.2.1 Navegación por el museo, Ir A, Búsqueda de Objetos

La funcionalidad es la misma que para un usuario visitante.

6.2.2 Cerrar Sesión

Sirve para cerrar la sesión iniciada por un investigador. Es útil porque elimina ciertas carpetas temporales que no son necesarias y que se crean cuando el investigador empaqueta algún objeto.

6.2.3 Subir OV

Permite subir un objeto virtual empaquetado que se encuentra en la máquina del usuario. Se abre un navegador de archivos que permite acceder hasta la carpeta comprimida que se quiere subir, figura 12. Cuando un objeto se sube se comprueba si está bien empaquetado y si es válido para pertenecer al museo. Para que el objeto sea válido, la carpeta debe tener un archivo manifiesto (imsmanifest.xml), un recurso llamado objetovirtual.xml donde se recogen los datos de las fichas y las dependencias entre ese objeto y otros del museo. En caso de que el objeto no sea válido, es decir no haya sido empaquetado con la herramienta que dispone el museo, se muestra un mensaje de error.

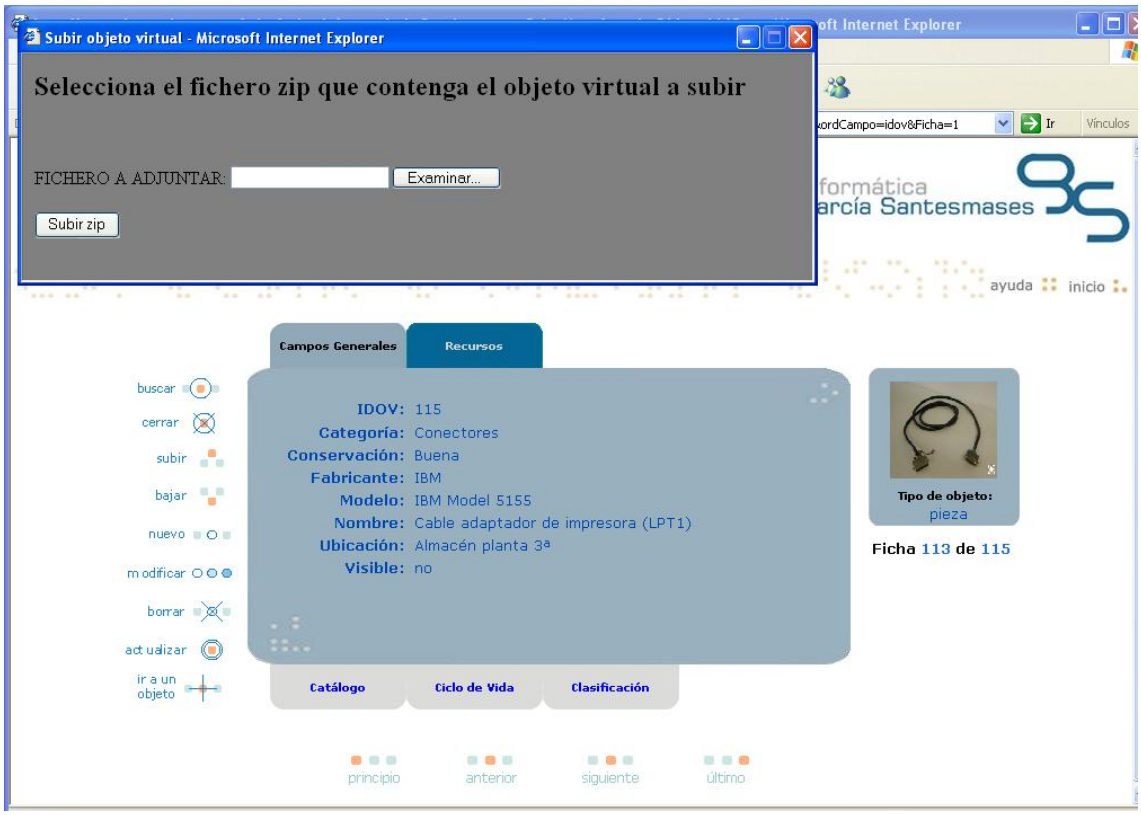

Figura 20: Subir OV.

6.2.4 Bajar OV

Permite bajar o exportar un objeto virtual del museo para que pueda ser utilizado en otros sistemas. Este objeto baja empaquetado según el estándar de IMS y se baja una carpeta comprimida que contiene los siguientes archivos:

- El archivo manifiesto (imsmanifest.xml).
- Los esquemas para validar este archivo xml (ims_xml.xsd, imsmd_v1p2p2.xsd e imscp_v1p1p3).
- Los recursos propios del objeto, entre los que se encuentra uno especial, objetovirtual.xml. El recurso objetovirtual.xml contiene todos los datos de todas las fichas del objeto.
- La DTD del objetovirtual.xml.
- Una hoja de estilo ObjetoVirtual.xsl que permite visualizar de una manera amigable el archivo objetovirtual.xml. Abriendo este archivo se tiene una visión del objeto en formato html. Este html contiene todos los datos de todas las fichas del objeto, así como enlaces a los recursos propios y a otros archivos objetovirtual.xsl de otros objetos virtuales de los que dependía el objeto principal.
- Todos los objetos que contienen dependencias con el objeto virtual que se descarga también son empaquetados.

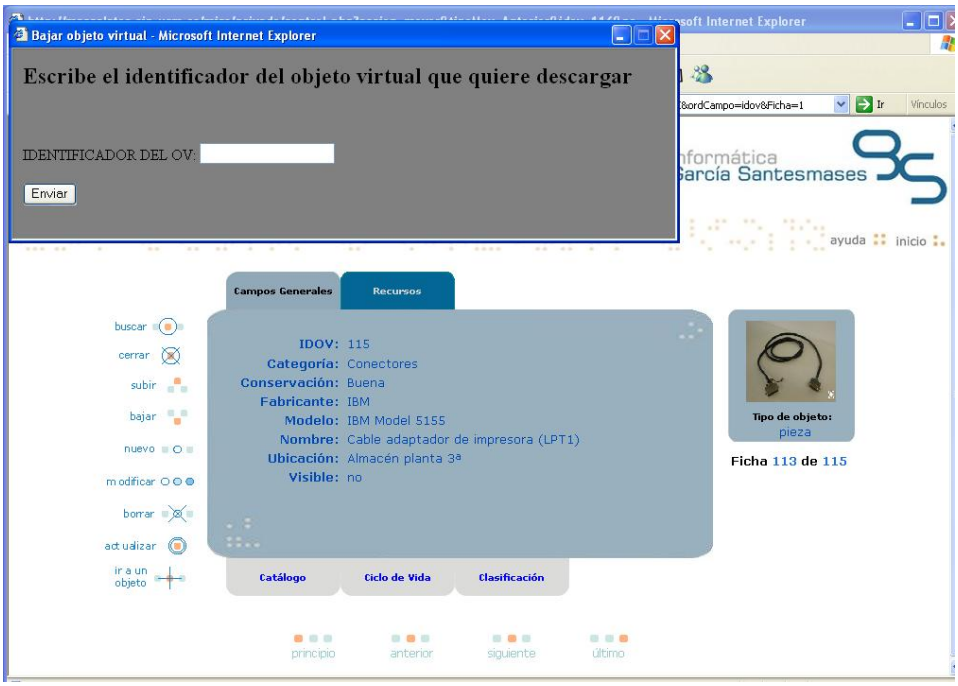

Figura 21. Bajar un objeto virtual del museo.

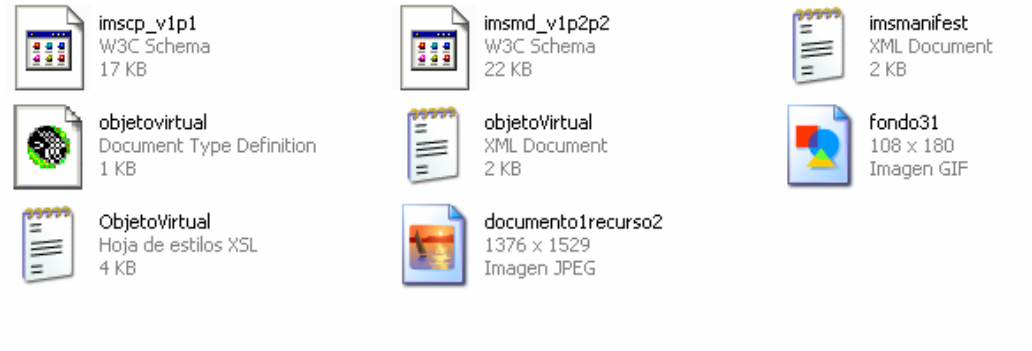

Figura 22: Contenido del objeto que se ha bajado del museo.

6.2.5 Nuevo OV

Con esta opción se permite crear un nuevo objeto virtual que podrá ser subido posteriormente al museo. Pulsando sobre este botón se accede a la herramienta de empaquetado que aparece en la Figura 23.

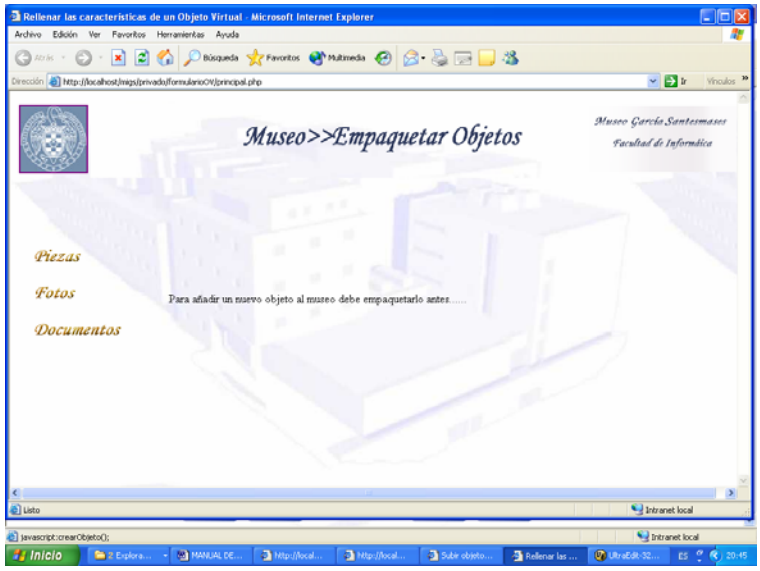

Figura 23: Ficha inicial para la creación de un objeto.

En el museo existen tres tipos de objetos: Piezas, Fotos y Documentos, por tanto lo primero que hay que hacer es seleccionar el tipo de objeto que se va a empaquetar. Dependiendo del objeto se rellenan unas fichas u otras.

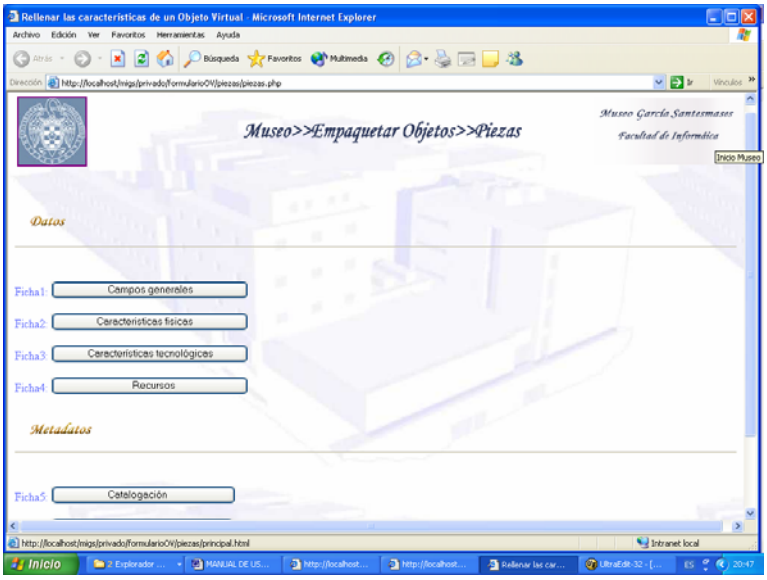

Figura 24: Página desde la que se accede a todas las fichas que se pueden rellenar para empaquetar una pieza.

Se encuentran diferenciadas entre las fichas de datos y las fichas de metadatos. Entre las de datos se encuentra la ficha Recursos, que es donde se añaden los recursos y las dependencias del objeto virtual.

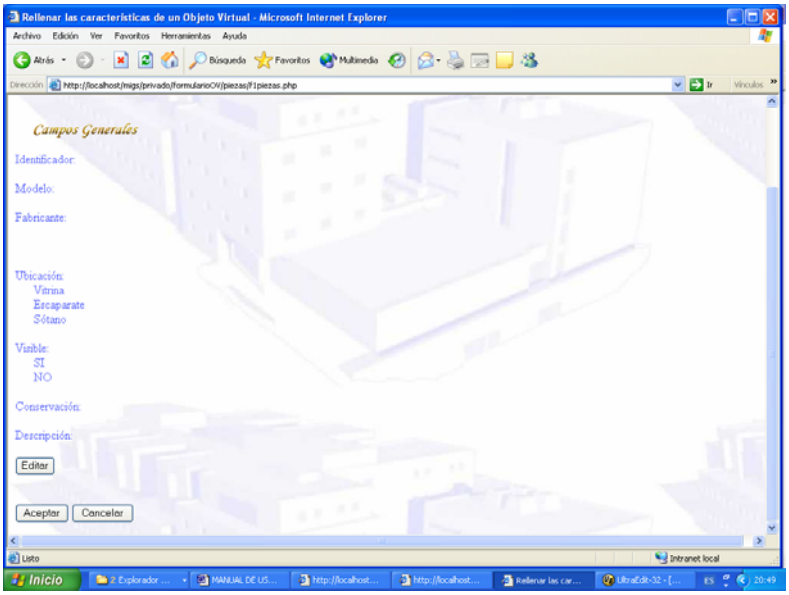

Figura 25: Campos de la ficha "Campos generales" de una pieza.

Pulsando sobre el botón editar de la Figura 25 se accede a la página donde se rellena los datos, Figuras 26a y 26b. Una vez introducidos los datos dichas fichas se vuelve a la Figura 25 que tendrá los campos rellenos. Si el usuario está conforme debe pulsar sobre el botón "Aceptar", todos los datos de la ficha quedan guardados y se vuelve a la página de la Figura 24 donde se puede volver a rellenar otra ficha si se desea. En caso de que el usuario no esté conforme con algún dato se puede modificar pulsando de nuevo sobre el botón "Editar".

Figura 26a: Formulario para rellenar los datos de una ficha.

Los campos que se ven en esta figura, son los que aparecen por defecto. El usuario investigador puede introducir otros atributos que considere oportunos para describir al nuevo objeto. Para ello, en el lado derecho, se pueden introducir pares de atributo valor, o solo introducir el nombre del atributo para darle valor en otro momento.

Los atributos por ficha son únicos, no se permite introducir un nuevo atributo si ya existe otro con el mismo nombre. La aplicación no permitiría la introducción y se avisa al investigador.

Algunos de los campos que aparecen por defecto, tienen un menú desplegable dinámico. Dinámico, ya que si el usuario no encuentra en ese menú, un valor para el atributo, se rellenaría el campo asociado "otra" en el que se introduciría un nuevo valor a añadir a ese menú desplegable. Para que se introduzca el nuevo valor en el menú, después de introducirlo en otra, se debe pulsar ACEPTAR.

Ese nuevo valor para ese atributo ya queda permanentemente reflejado, ya que si se despliega ese menú asociado a ese atributo posteriormente para modificar o crear otro ov de la misma categoría, aparece ese valor para poder elegirlo.

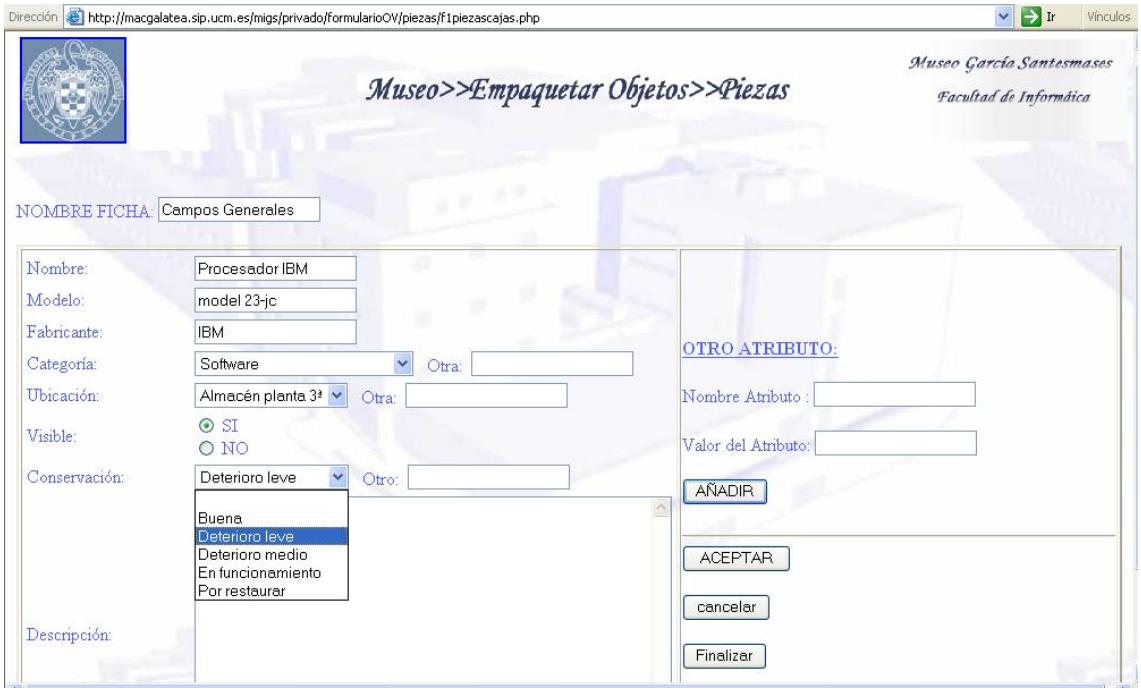

Figura 26b: Formulario relleno de una ficha.

Una vez de acuerdo con los datos se debe pulsar en el botón **ACEPTAR** para guardar los cambios y si se está conforme se debe pulsar en el botón **FINALIZAR**.

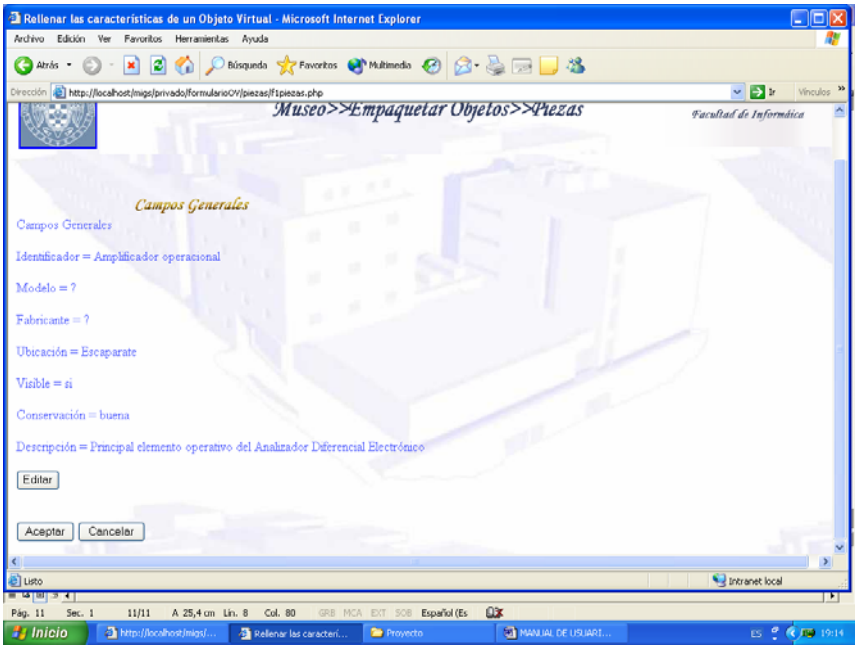

Figura 27: Página donde se muestran los datos introducidos.

Una vez pulsado en el botón **Finalizar** se llega a la página de la Figura 27, donde se muestran los datos introducidos por el usuario, en caso de que no esté conforme se pueden modificar volviendo a hacer clic en el botón **Editar** que llevará hasta la página de la Figura 26b. En caso de que el usuario esté conforme con los datos introducidos debe pulsar en el botón **Aceptar** y volverá a la página de la Figura 24 para seguir introduciendo nuevas fichas al objeto virtual.

Una vez el usuario haya rellenado todas las fichas que deseé debe introducir un nombre para el nombre del paquete que se va a crear en la caja de texto de la Figura 28 y pulsar sobre el botón **"Crear Objeto Virtual"**.

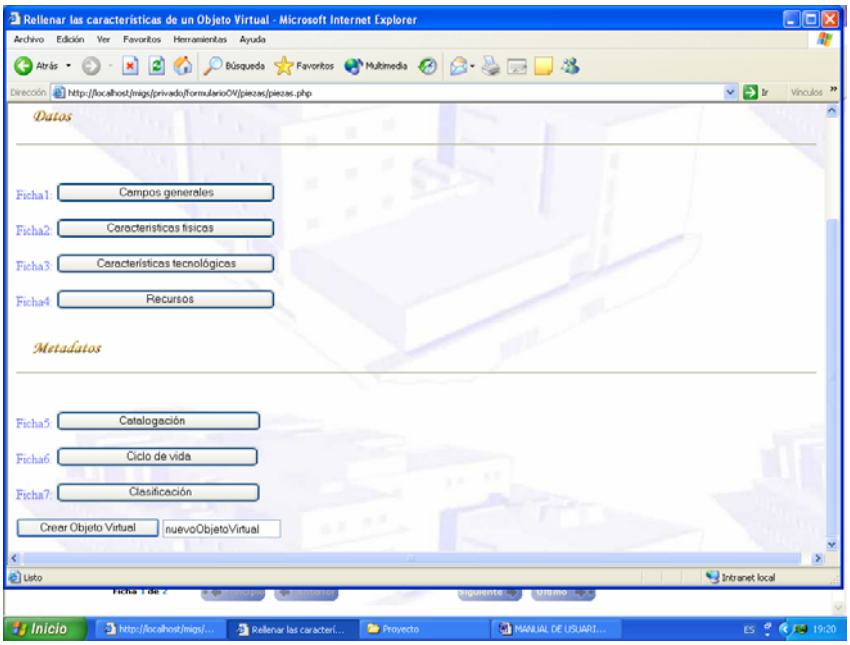

Figura 28: Página desde donde se acceden a todas las fichas del objeto.

Si pulsamos el botón **"Crear Objeto Virtual"** se llega hasta una página de información que indica que el objetoVirtual.xml ha sido creado (recurso imprescindible

para el paquete) y un botón **"Crear Manifiesto".** Pulsando ese botón, se crea el manifiesto y da la opción de guardar ese ov en el ordenador, para poder posteriormente subirlo al museo o transportarlo a otro sistema. Una vez creado el objeto se puede subir al museo.

6.2.6 Modificar OV

Como ya se ha comentado anteriormente, un objeto del museo sólo puede ser modificado por los investigadores que tengan permisos de edición sobre él. Inicialmente cada investigador podrá modificar los objetos creados a no ser que el administrador les quite esos privilegios.

El administrador del museo virtual de informática, podrá modificar cualquier objeto virtual almacenado en el museo.

Figura 29: Modificación de un objeto.

Posteriormente, se pulsaría en Piezas (en este caso) y se mostraría la Figura 24. Desde aquí la funcionalidad es muy similar a de "crear OV".

6.2.7 Borrar OV

Al igual que la opción anterior, sólo los investigadores con permisos sobre un objeto o el administrador pueden eliminar un objeto del museo.

Sólo se puede borrar objetos de los que no depende ningún otro objeto. Con esto se evitan los posibles errores en la navegación entre objetos relacionados.

6.2.8 Actualizar

Esta acción refresca el museo, para hacer visibles los últimos cambios efectuados. Al actualizar se queda visualizando el mismo objeto virtual donde se estaba antes de dar a actualizar.

7 Manual de Instalación

En esta sección se expondrá como realizar una instalación de la aplicación MIGS 1.0 en un entorno Windows, así como la de todos los prerrequisitos que son necesarios para su correcto funcionamiento.

Los servidores MAC suelen tener todos los prerrequisitos preinstalados y por ello no se realiza una guía sobre ellos. Aun así, esta guía sirve para realizar la configuración de los mismos.

7.1 Prerrequisitos

Antes de comenzar con la instalación de MIGS 1.0 será necesario tener instalados el servidor Web Apache, PHP y una Base de Datos MySQL.

Las versiones recomendadas para la instalación son:

- Servidor Web Apache 1.3.31
- PHP 4.3.3
- Base de Datos MySQL 4.0.17

Versiones más recientes de Apache, PHP y MySQL han dado problemas en el funcionamiento de MIGS y no son recomendadas para su funcionamiento. Con las versiones que aquí se muestran MIGS 1.0 ha sido testeado en su totalidad.

Es este manual se realizará una instalación de los prerrequisitos en la carpeta base “c:\migs” evitando en todo lo posible los espacios en directorios.

7.1.1 Servidor Apache

Se puede descargar en http://httpd.apache.org/download.cgi.

Haga doble clic en el ejecutable apache_1.3.31-win32-x86-no_src.exe.

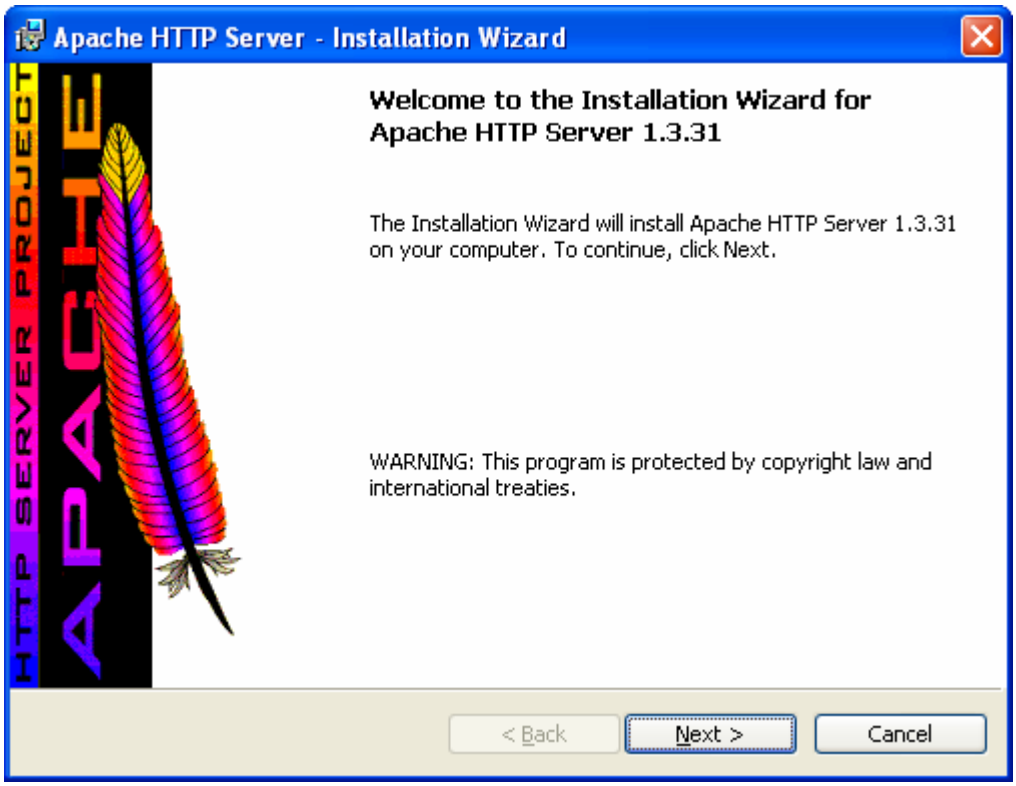

Figura 30: Instalación de Apache.

Especifique tanto en el dominio de red como en el nombre del servidor el valor "127.0.0.1". En el campo e-mail especifique alguno conocido, aunque no es estrictamente necesario.

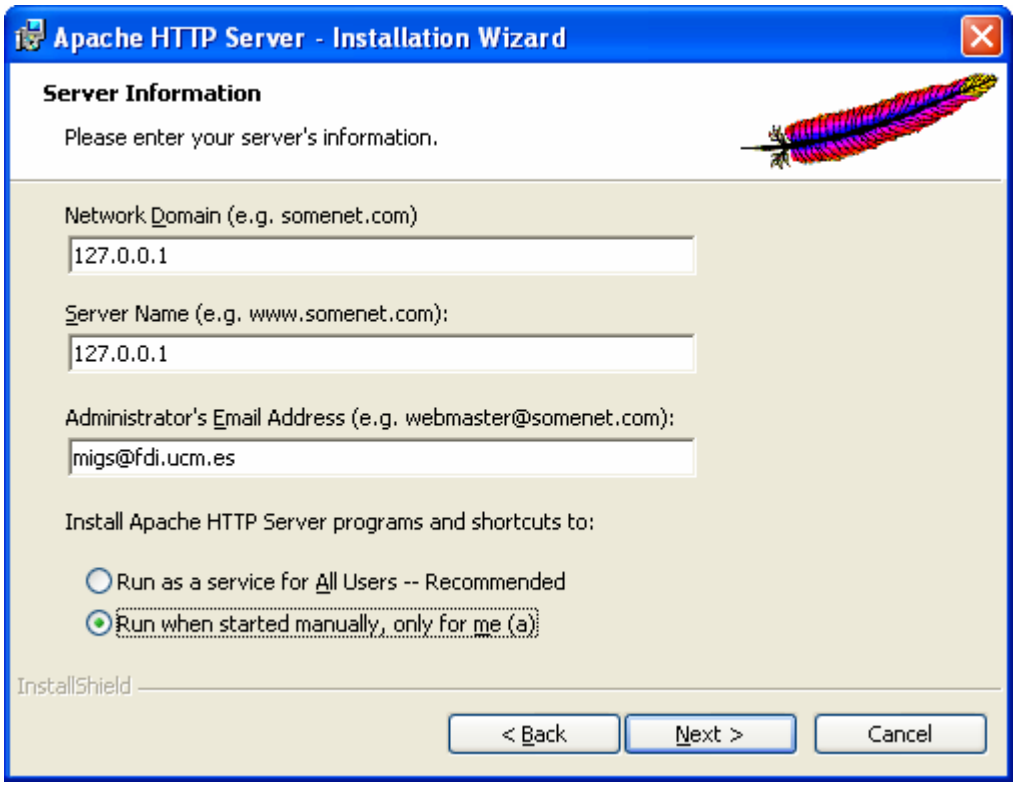

Figura 31: Configuración de nuestra red.

Seleccione la carpeta de instalación. Es recomendable que no contenga espacios para evitar futuros problemas.

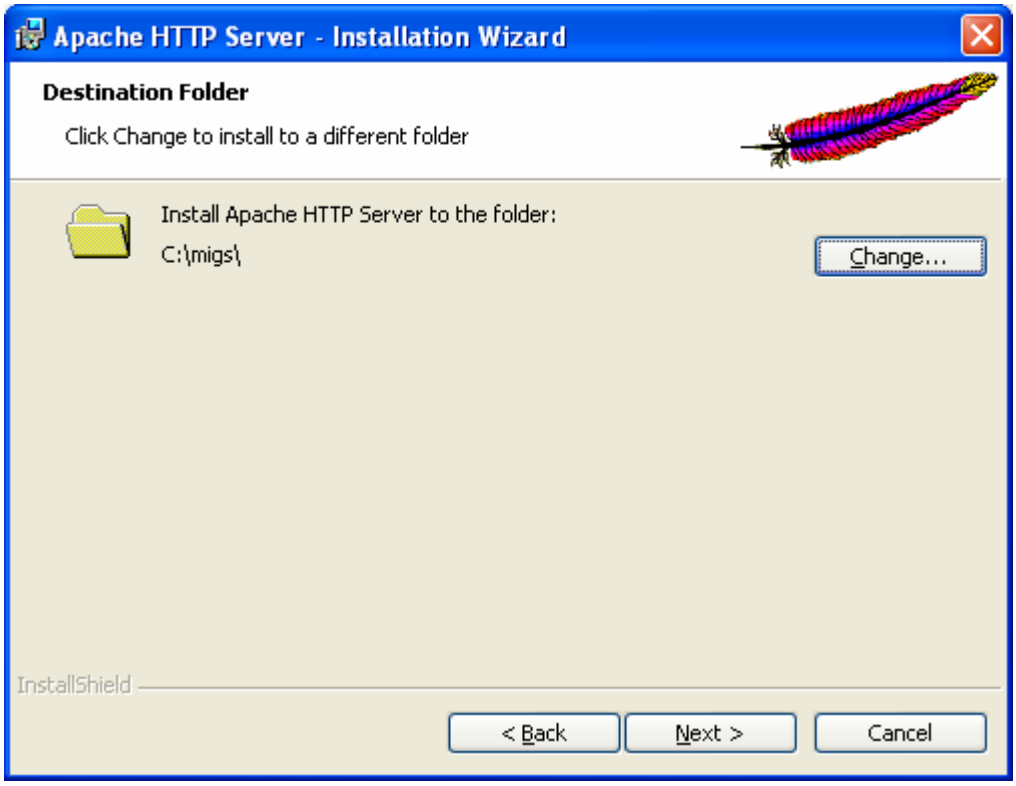

Figura 32: Directorio de instalación.

Continúe la instalación hasta que concluya.

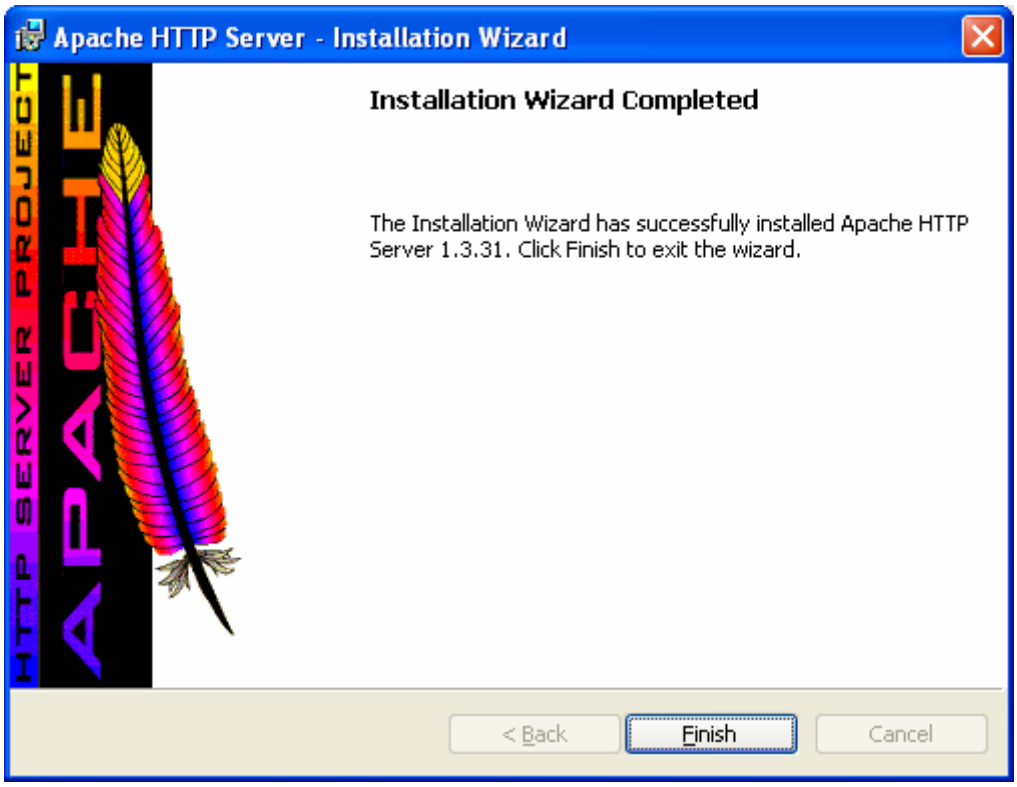

Figura 33: Instalación completada

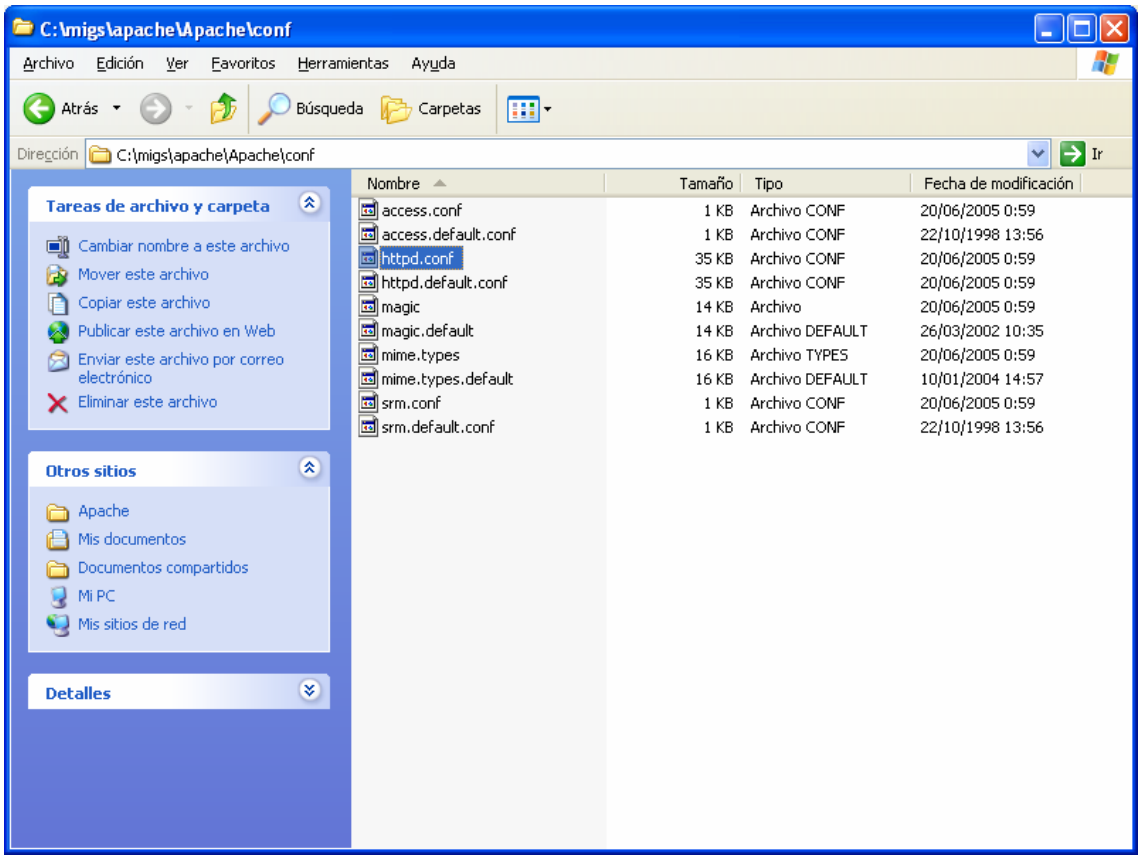

Figura 34: Fichero de configuración del servidor.

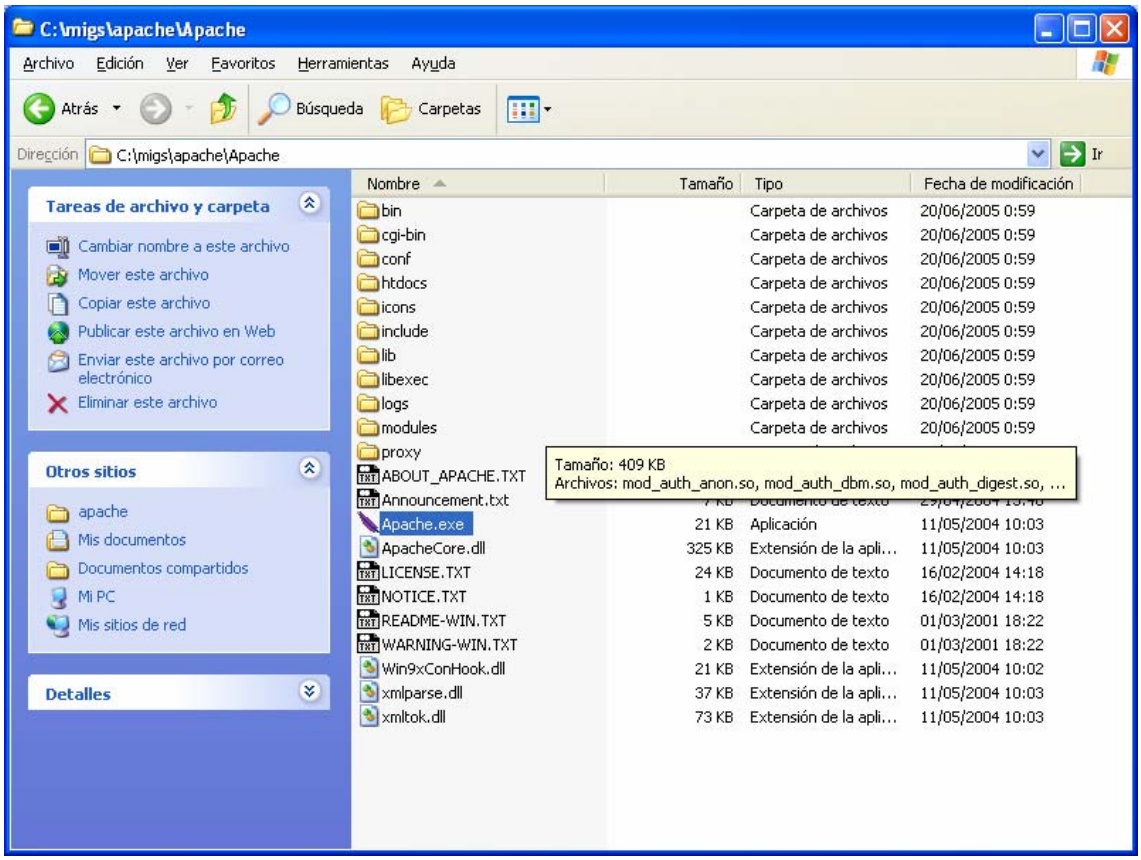

Figura 35: Archivo de iniciación de servidor Apache.

7.1.2 PHP 4.3.3

Se puede descargar en http://www.php.net/ en la sección “downloads”. Serán necesarios tanto los binarios de la versión 4.3.3 como las extensiones.

Crearemos una carpeta para descomprimir los binarios de PHP. Es recomendable una ruta sin espacios. Ej: c:\migs\PHP

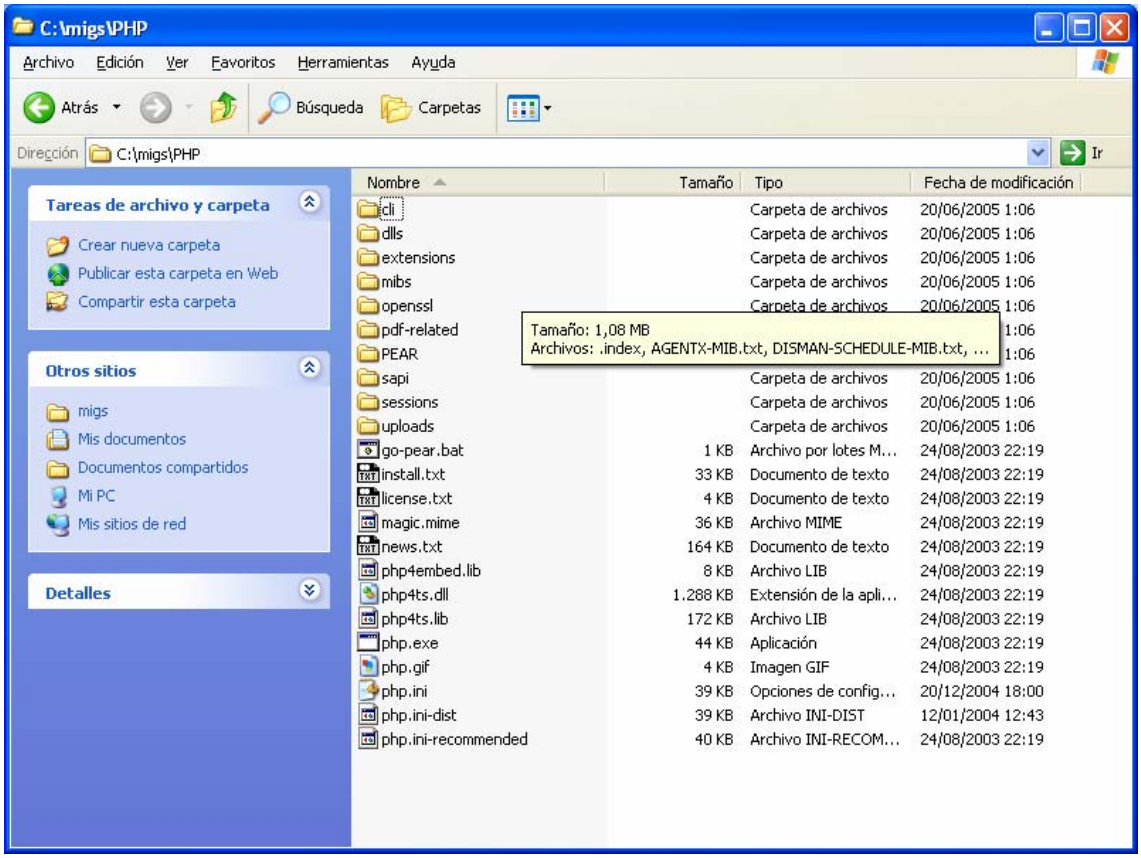

Figura 36: Contenido del directorio PHP

En esa misma carpeta crearemos un directorio “extensions” donde descomprimiremos las extensiones, “uploads” para la subida de ficheros y “sessions” para las sesiones.

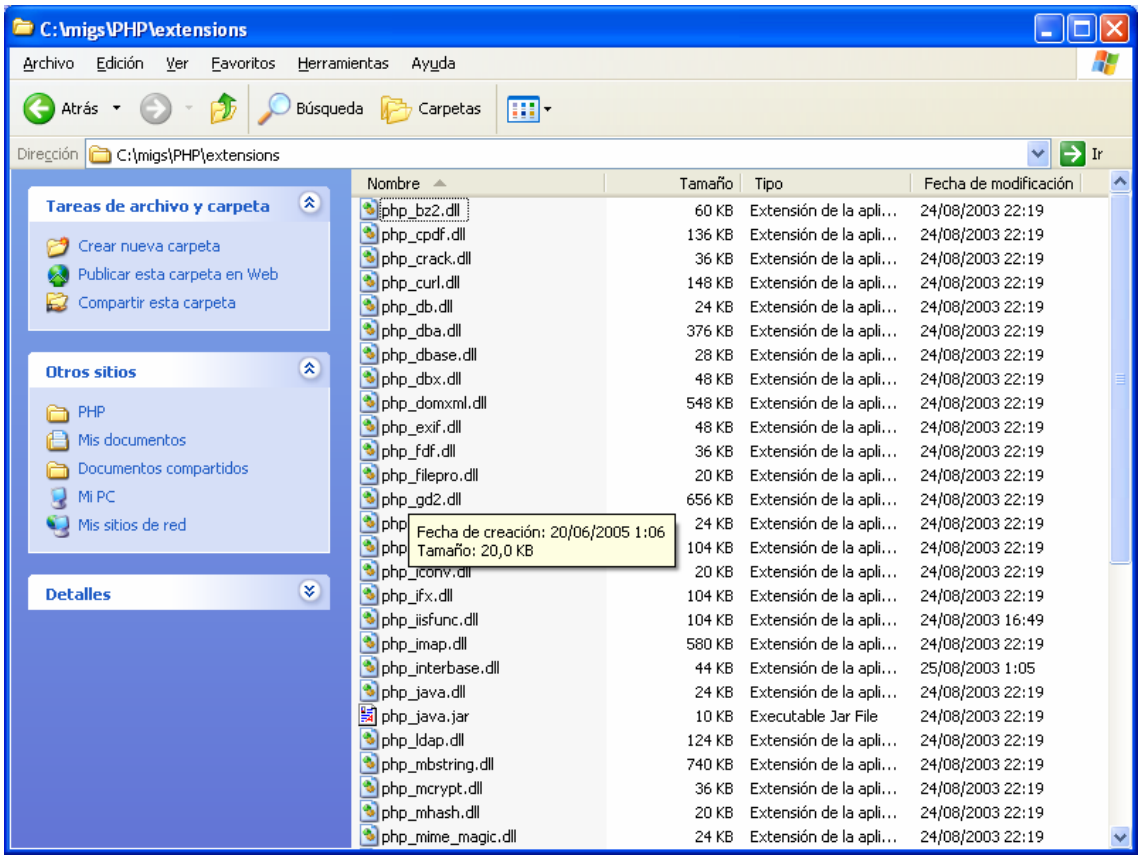

Figura 37: Extensiones de PHP.

Para el correcto funcionamiento de PHP es necesario o que tanto su directorio base como su directorio “dlls” se encuentren en el path del sistema o, en su defecto, se copie tanto el archivo “php4ts.dll” como el contenido de la carpeta “dlls” en el directorio “c:\windows\system32”

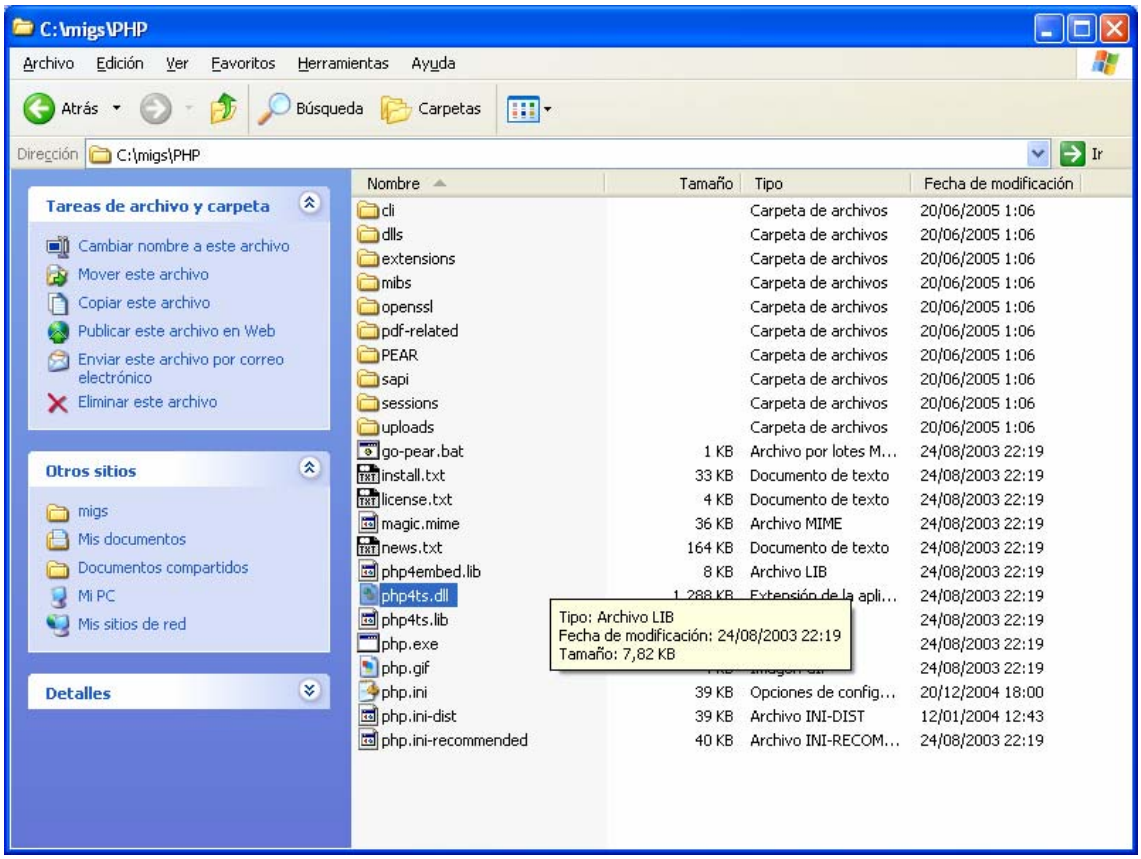

Figura 38: dll principal de PHP.

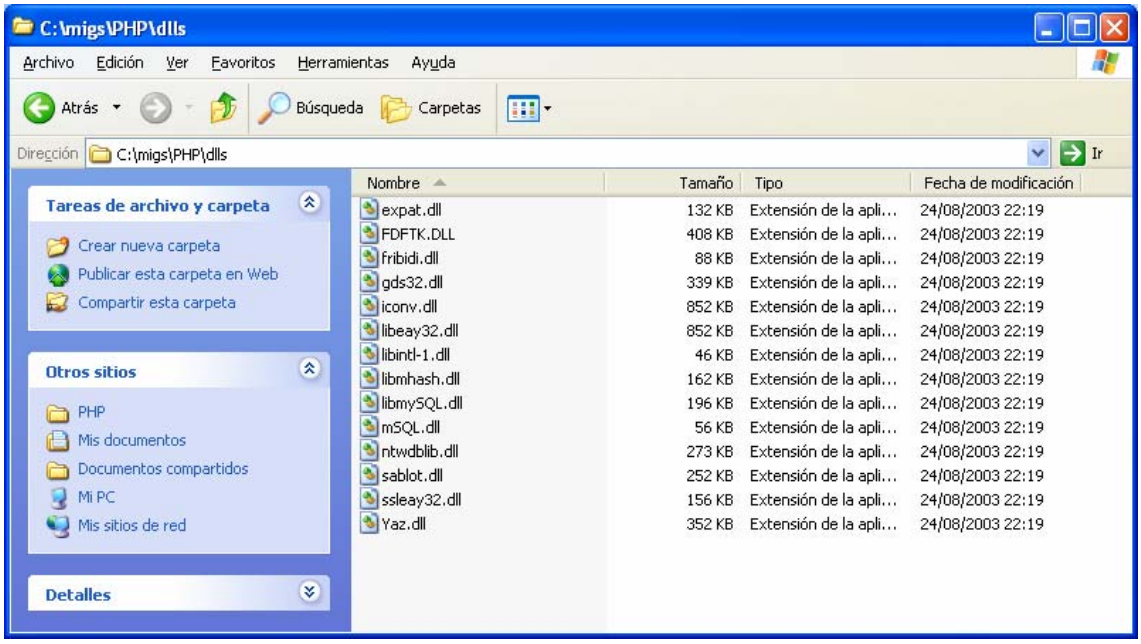

Figura 39: librerias de enlace dinámico secundarias de PHP.

Ahora procederemos a la configuración del archivo php.ini. Seleccione el recomendado del directorio base de PHP, renómbrelo a “php.ini”, cópielo a “c:\windows” y realice los cambios que se muestran a continuación. Si en la instalación se ha seleccionado algún directorio diferente a los mostrados en esta guía será necesario realizar los cambios pertinentes.

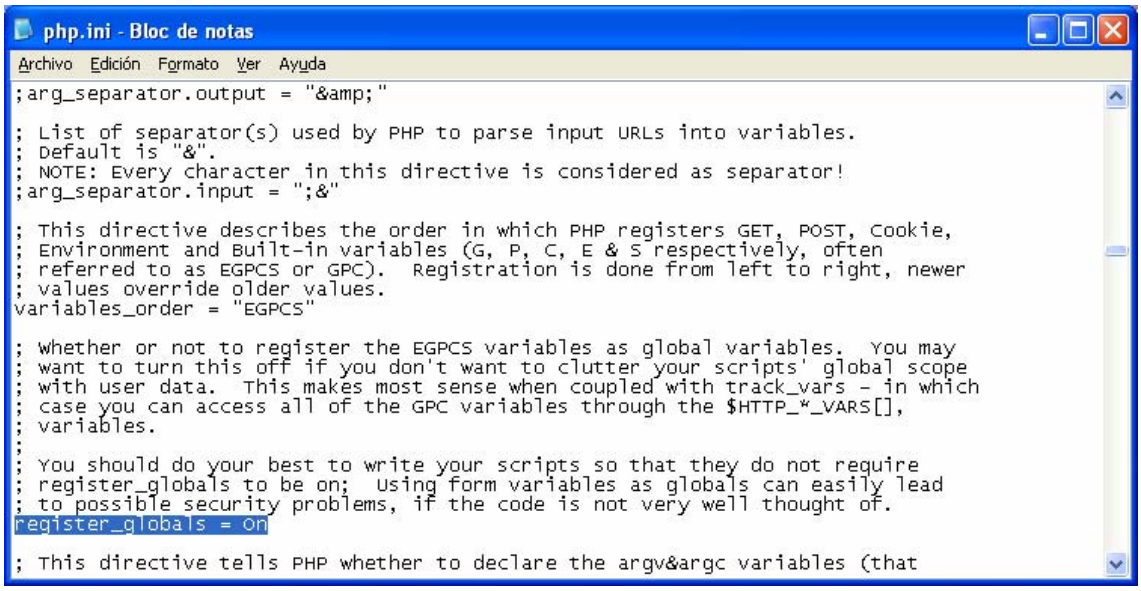

Figura 40: Permitir variables globales.

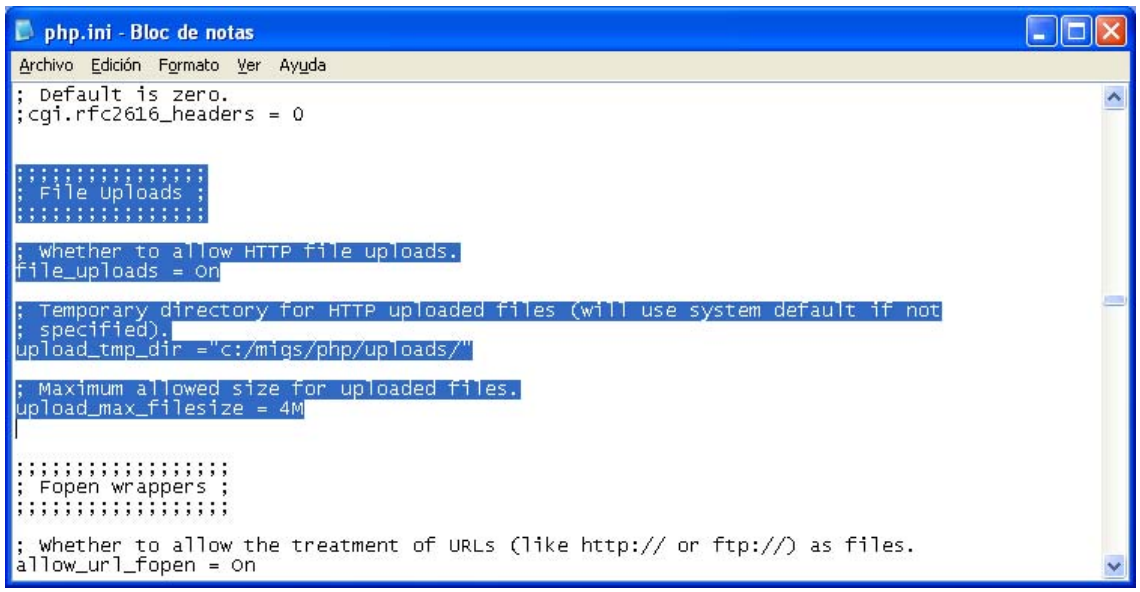

Figura 41: Configuración de la subida de ficheros

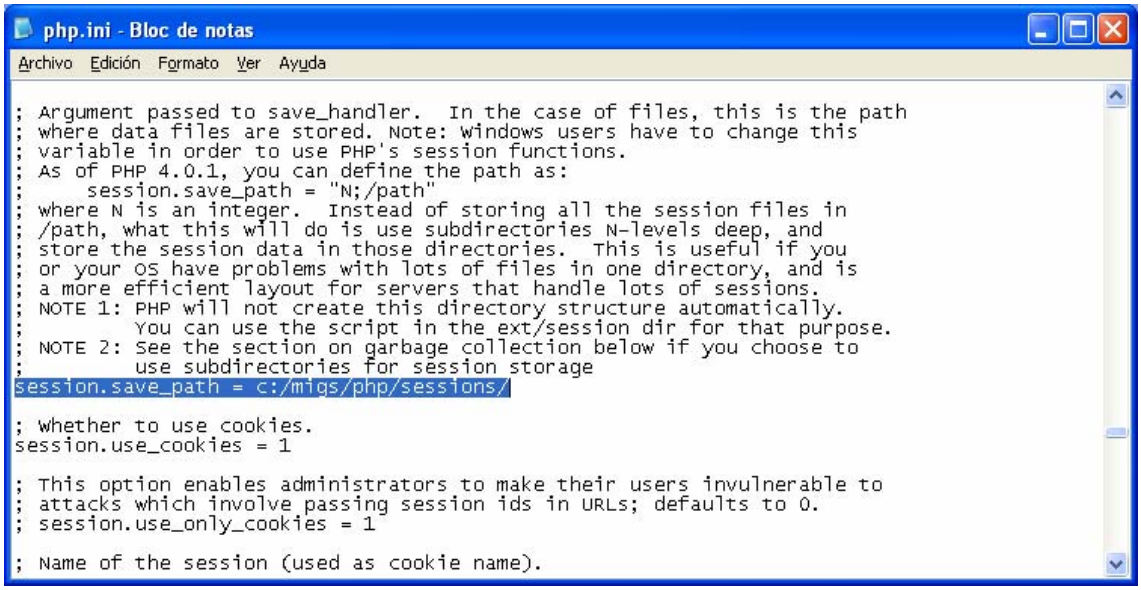

Figura 42: Configuración del directorio de las sesiones.

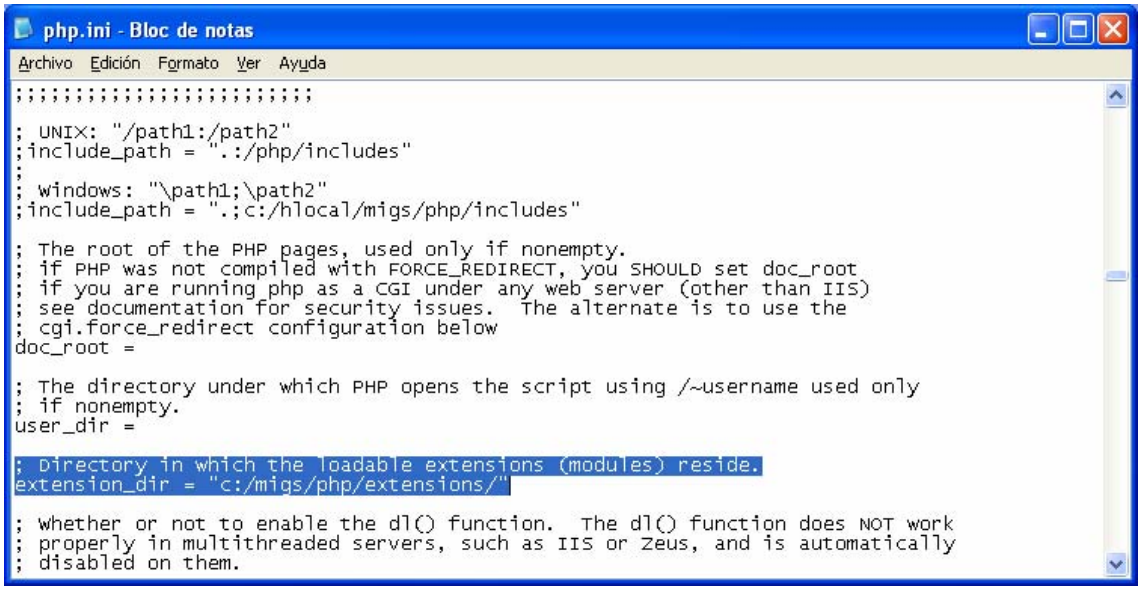

Figura 43: Directorio de las importaciones de PHP

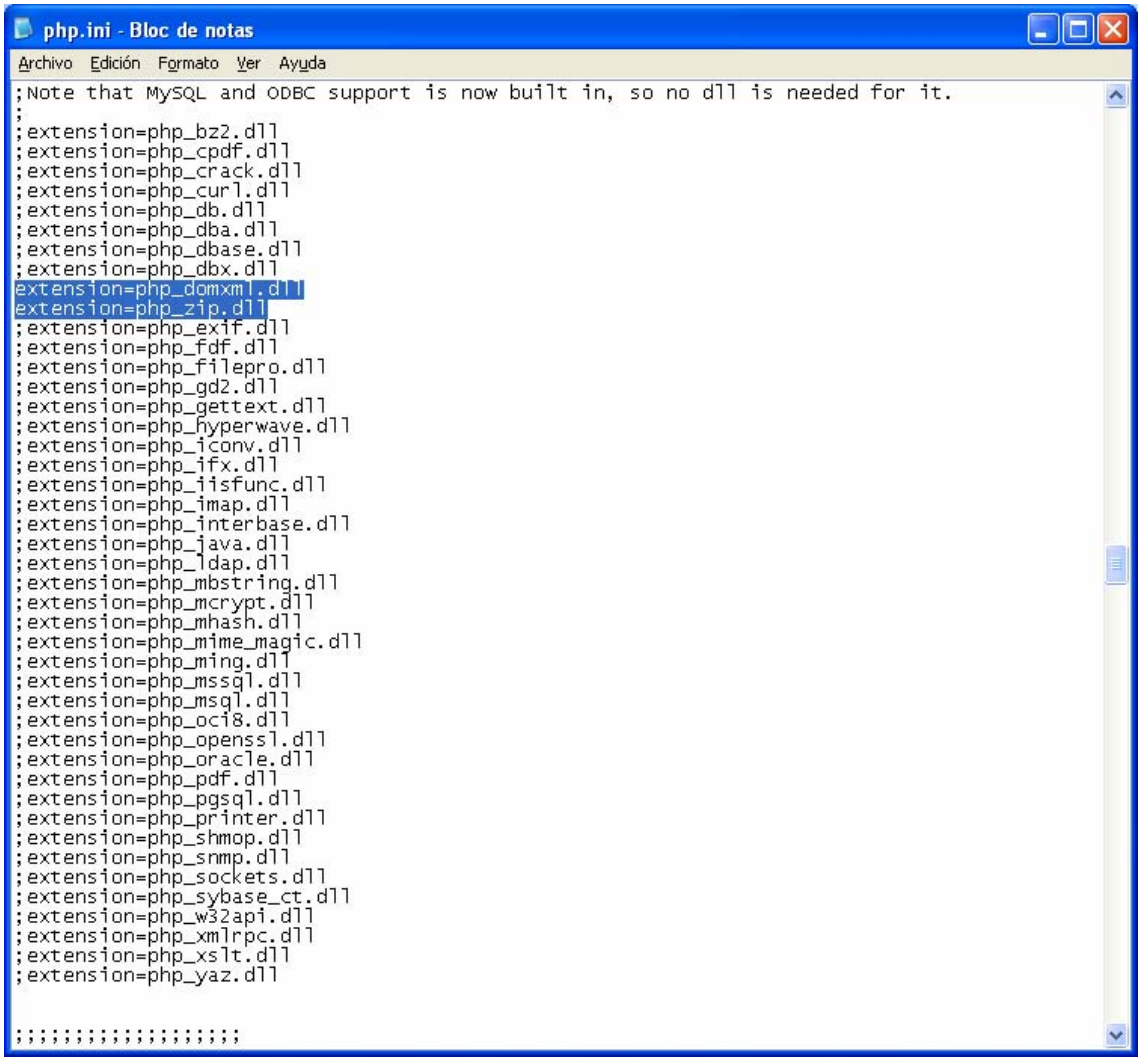

Figura 44: Importaciones necesarios en MIGS.

Guardaremos los cambios al salir.

El siguiente paso es configurar el servidor Apache para que trabaje conjuntamente con PHP.

Para ello editaremos el fichero de configuración del servidor “http.conf” ubicado en el directorio “conf” de la carpeta base del servidor y se realizarán los cambios que se muestran a continuación.

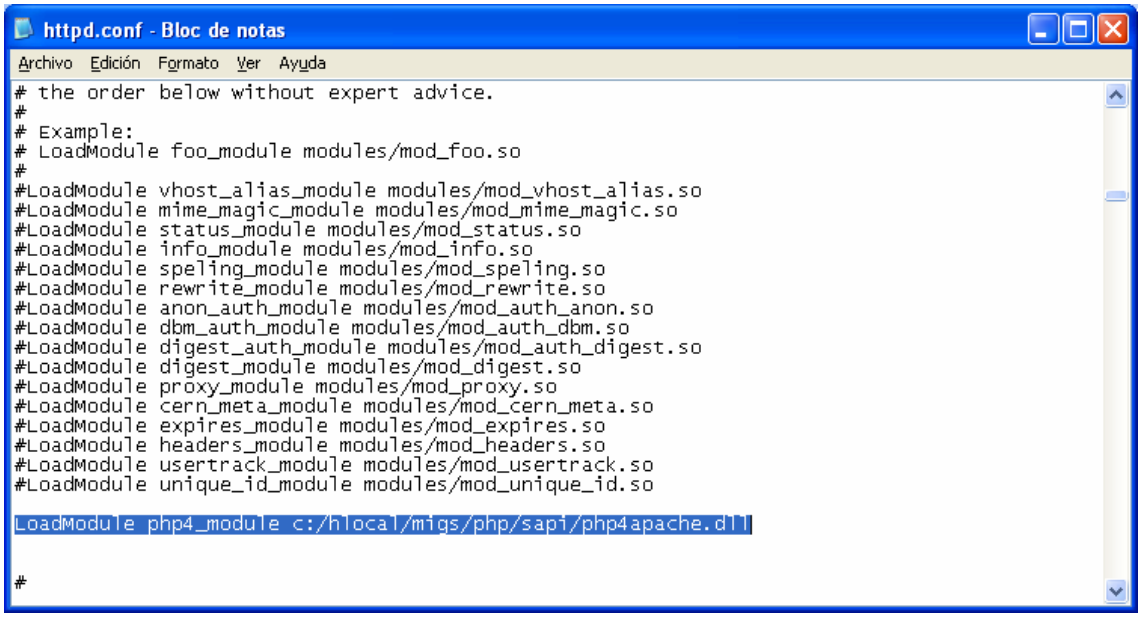

Figura 45a: Enlace PHP-Apache.

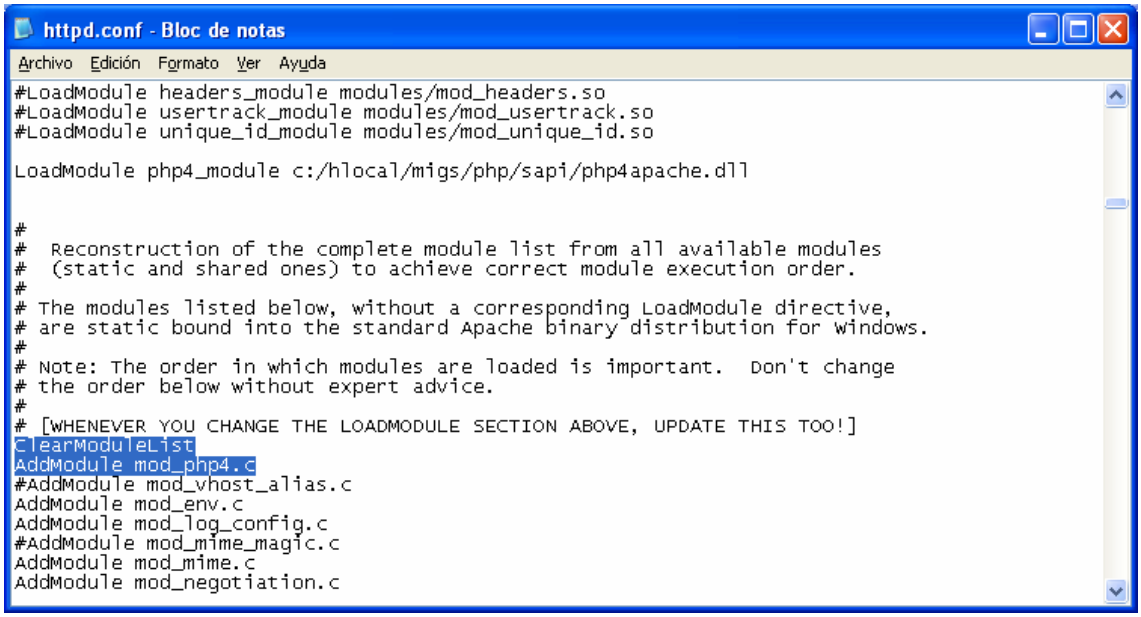

Figura 45b: Enlace PHP-Apache.

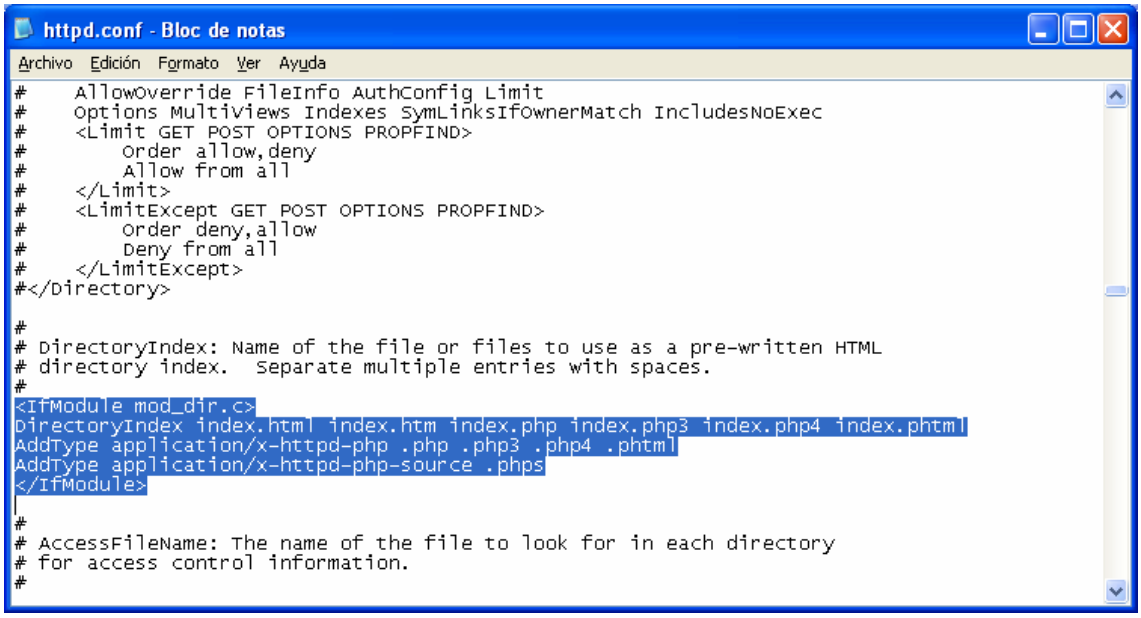

Figura 45c: Enlace PHP-Apache.

El siguiente paso es comprobar que la configuración ha sido realizada correctamente.

Para ello crearemos un archivo llamado “info.php” en la carpeta “htdocs” del directorio base de Apache con el siguiente contenido:

```
<?
phpinfo();
?>
```

Arrancaremos el servidor, abriremos un navegador y probaremos la URL http://localhost/info.php. Si todo ha ido correctamente se mostrará una pantalla con la configuración de PHP.

7.1.3 Base de Datos MySQL

Ejecutaremos el archivo de instalación de MySQL. En caso de no disponer de los binarios se pueden descargar en http://www.mysql.com/

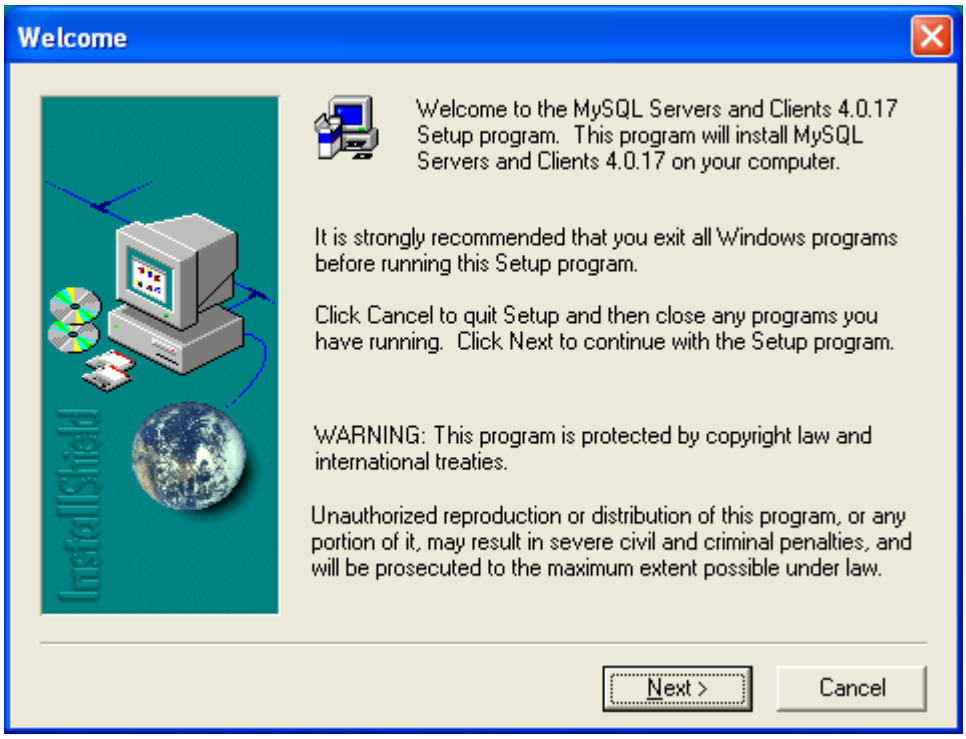

Figura 46: Instalación MySQ.

Indicaremos el directorio deseado para la instalación. Se recomienda que no contenga espacios para evitar problemas futuros.

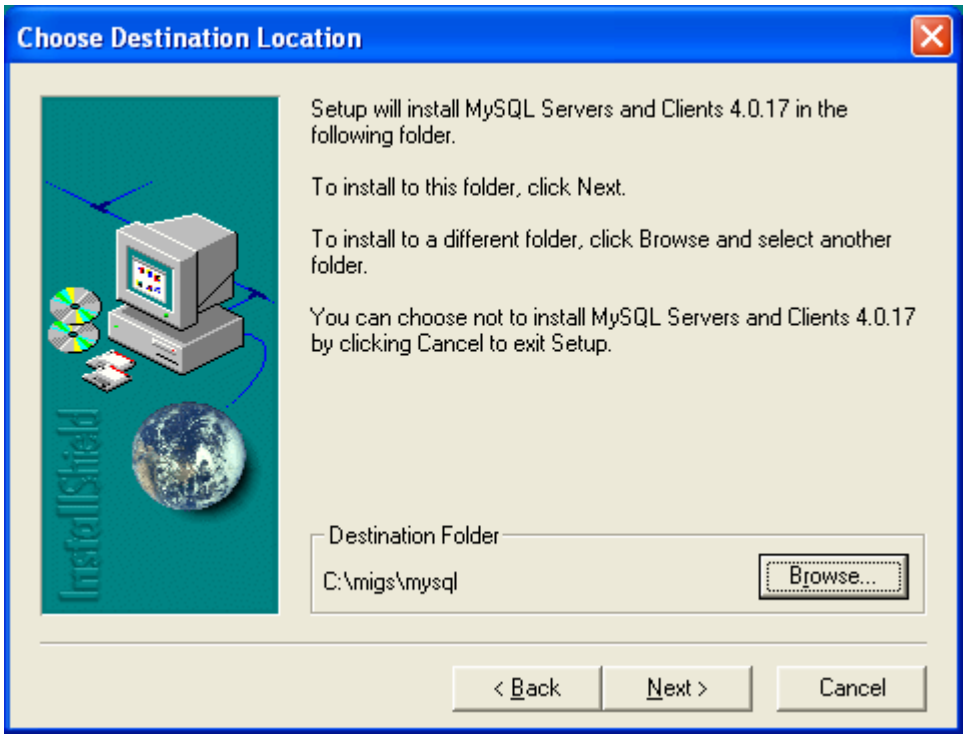

Figura 47: Directorio de instalación de MySQL.

Llevaremos a cabo la instalación.

Una vez concluida ejecutaremos “winmysqladmin.exe”, ubicado en el directorio “bin” de la carpeta que se seleccionó durante la instalación. La imagen de un semáforo en el menú inicio reflejará que la instalación se ha realizado satisfactoriamente.

Como herramienta adicional para la administración de la base de datos instalaremos el MySQL Control Center, que aporta un interfaz fácil e intuitivo para cualquier usuario.

Figura 48: Instalación MySQL Control Center.

Indicaremos la ubicación de la instalación

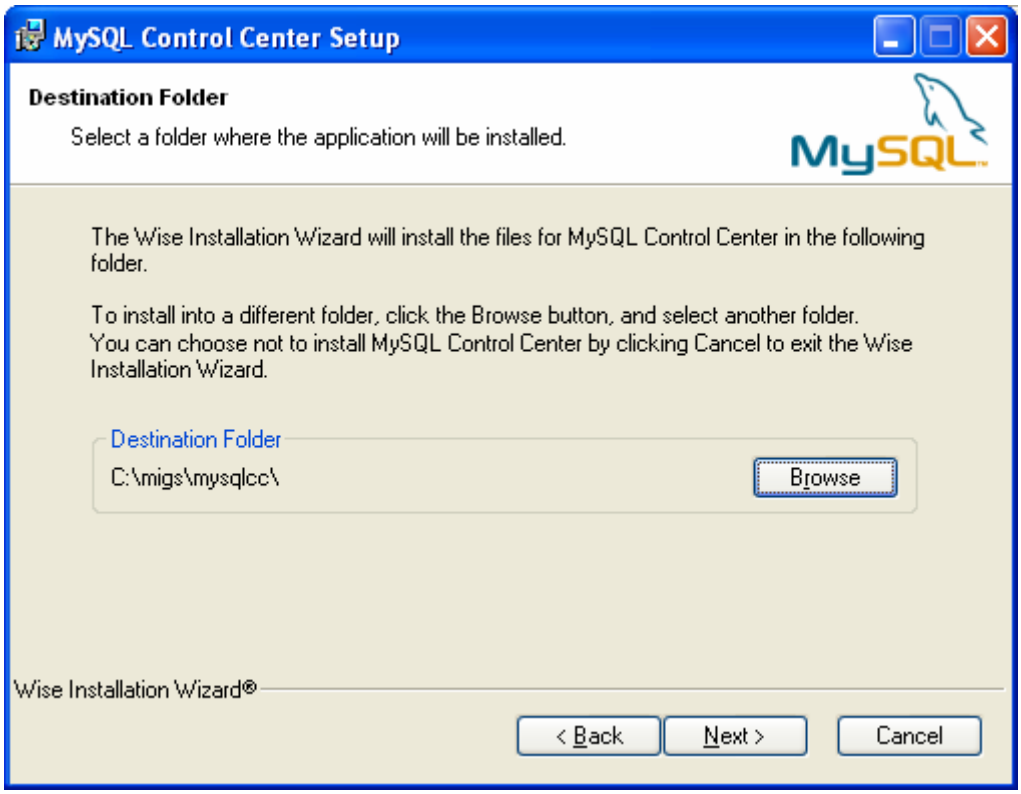

Figura 49: Directorio de instalación MySQL Control Center.

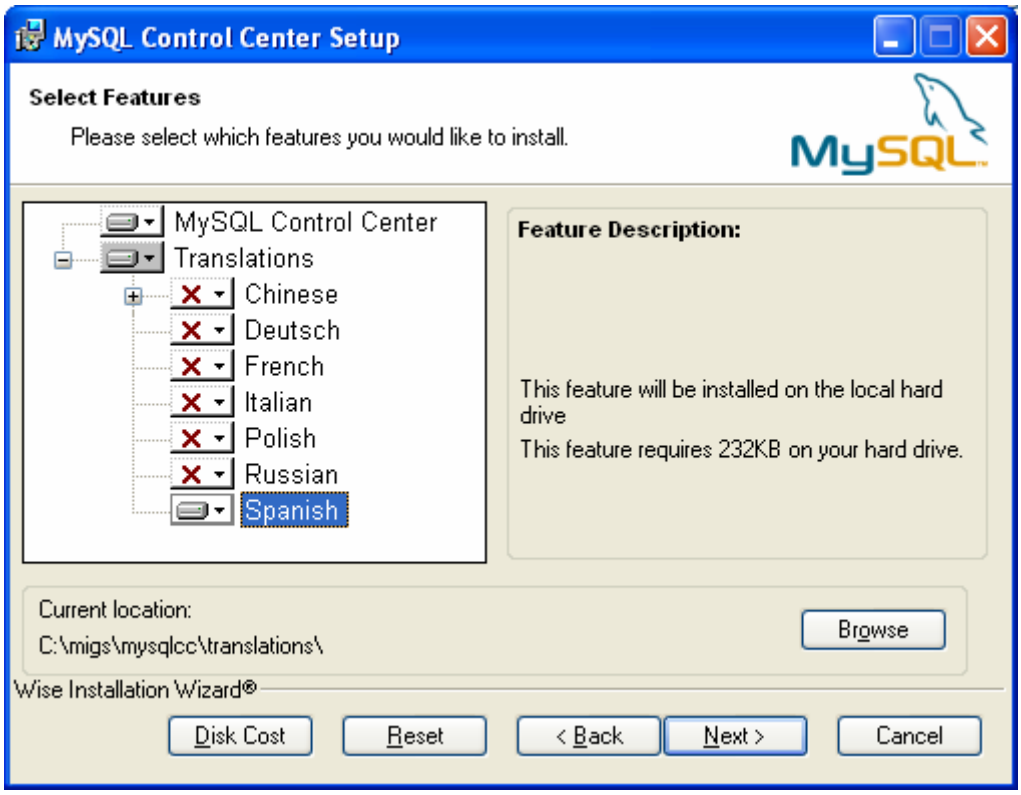

Figura 50: Configuración del idioma.

Continuaremos hasta el final de la instalación.

7.2 Instalando MIGS 1.0

Una vez instalados todos los prerrequisitos procederemos a la instalación del museo virtual.

7.2.1 Aplicación Web

Sitúese en la carpeta base de Apache y entre al directorio “htdocs”. Cree una carpeta llamada “migs” y copie el contenido da la aplicación MIGS en ella.

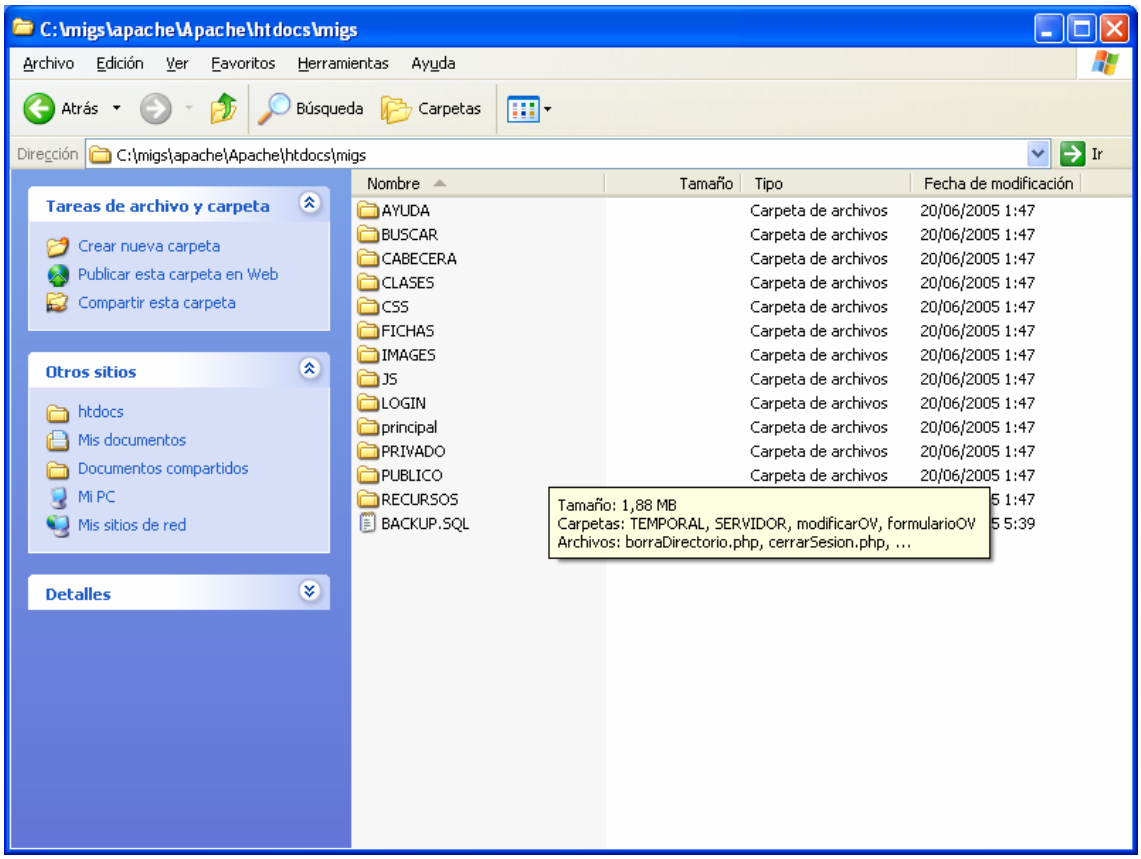

Figura 51: Creación de la aplicación “migs”.

7.2.2 Base de Datos

Abriremos el *MySQL control center* y crearemos una nueva Base de datos llamada “migs”.

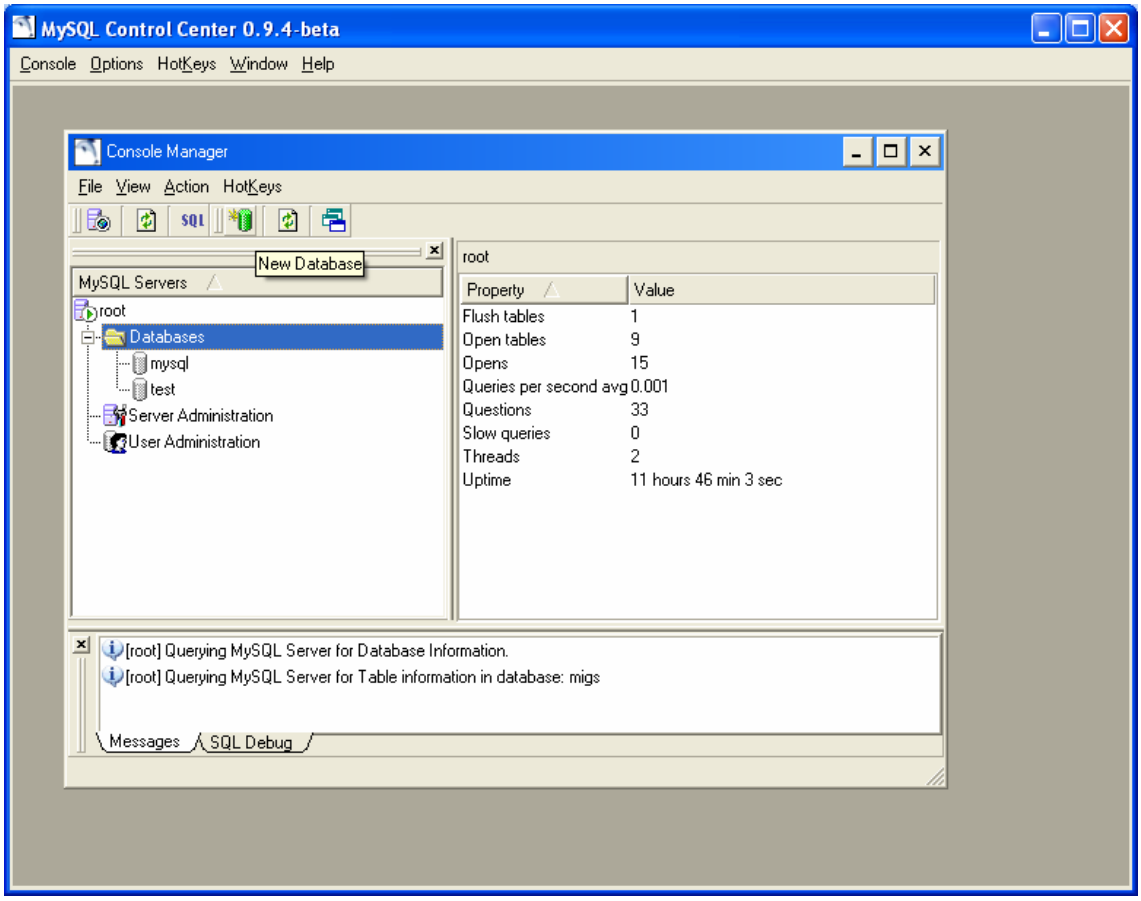

Figura 52: Pantalla principal mySQL Control Center.

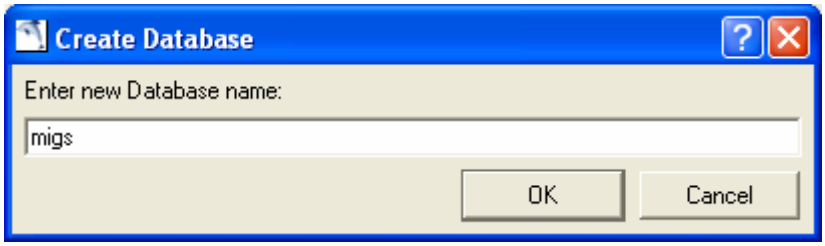

Figura 53: Creación de la base de datos “migs”.

Haga doble clic en la nueva Base de Datos creada y abra una línea de comandos SQL.

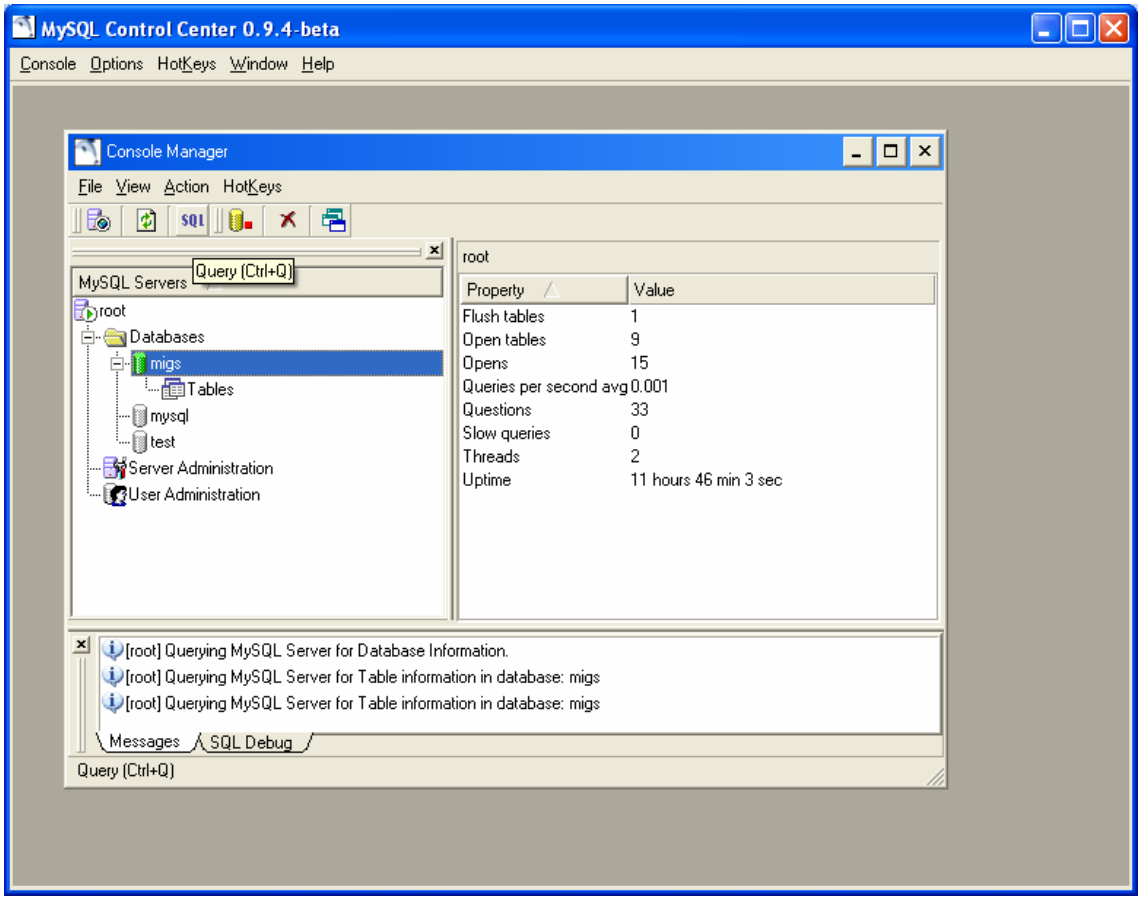

Figura 54: Conexión a la base de datos.

Ejecute el código SQL que se encuentra en el archivo “BACKUP.SQL” de la carpeta base de MIGS para la creación de las tablas vacías y de un usuario administrador.

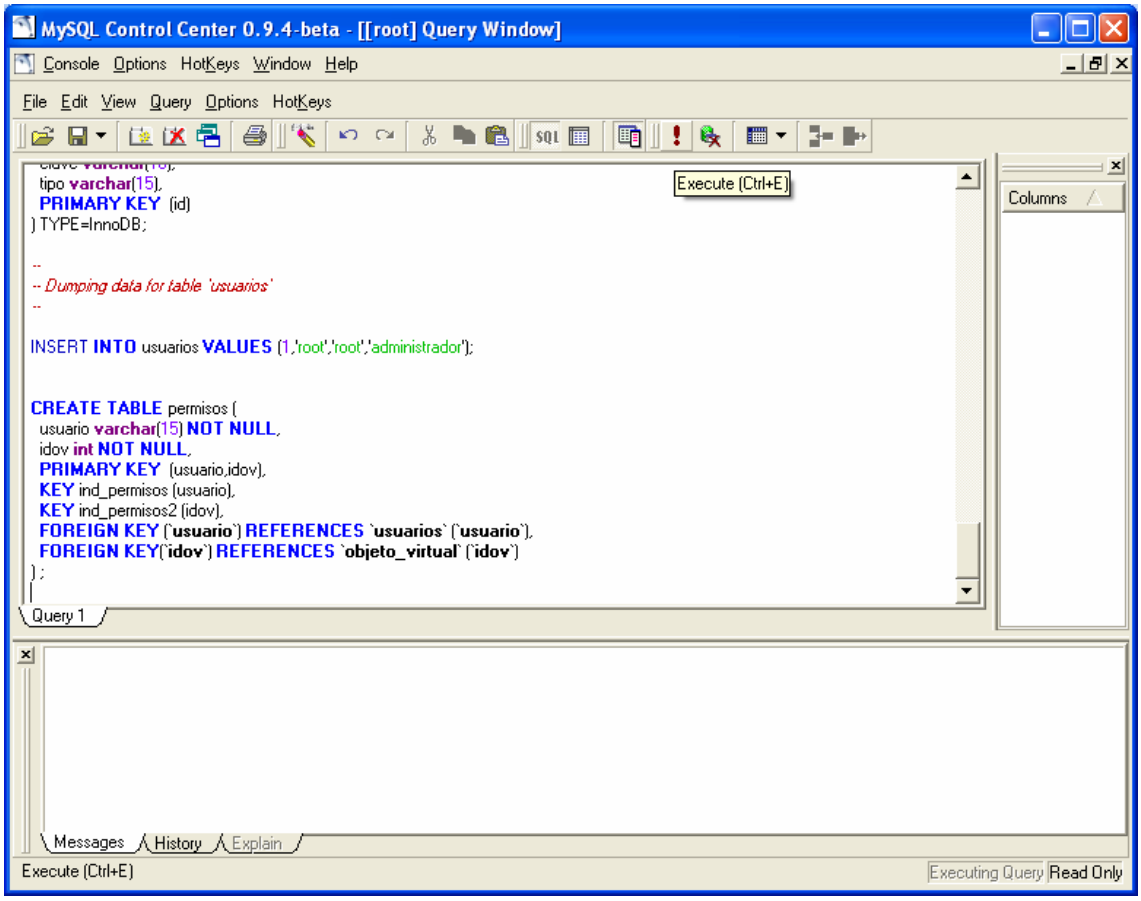

Figura 55: Ejecución del script de MIGS.

7.2.3 Comprobando el Correcto Funcionamiento

En este apartado realizaremos una navegación por MIGS probando las funcionalidades más conflictivas. Si durante la misma surgen problemas acuda a la sección “Problemas y Soluciones” y revise los pasos de la instalación.

Arranque tanto el servidor Apache como la base de datos MySQL. Abra un navegador e introduzca la siguiente dirección:

http://localhost/migs/principal/principal.html

Debería encontrar la página principal de MIGS. Entre a la sección de investigadores he introduzca su login y password. Ej: root, root.

Actualmente se encuentra visitando los objetos del museo. Si no hay ningún objeto introducido tendrá la opción de crearlo. Cree un nuevo *Objeto Virtual* de la categoría *Pieza*.

Rellene al menos un dato de cada ficha he introduzca al menos tres recursos propios al nuevo objeto, el primero de ellos una imagen, el segundo un archivo pdf y por último un archivo html simple.

Finalice la creación del objeto y visualícelo en el museo.

Edítelo y compruebe los cambios de la edición.

En la pantalla de navegación pulse la opción “Bajar” para descargar un zip con el *Objeto Virtual* comprimido.

Pulse “Subir” y seleccione el archivo que descargó en el paso anterior. Se creará en el museo un nuevo objeto similar al ya existente.

7.3 Problemas Y Soluciones

7.3.1 El Servidor Web no Arranca

Posiblemente el puerto 80 esté ocupado por otra aplicación.

Edite el fichero de configuración de Apache http.conf y cambie la línea correspondiente al puerto por otro puerto libre. Ej.: 8080, 8085.

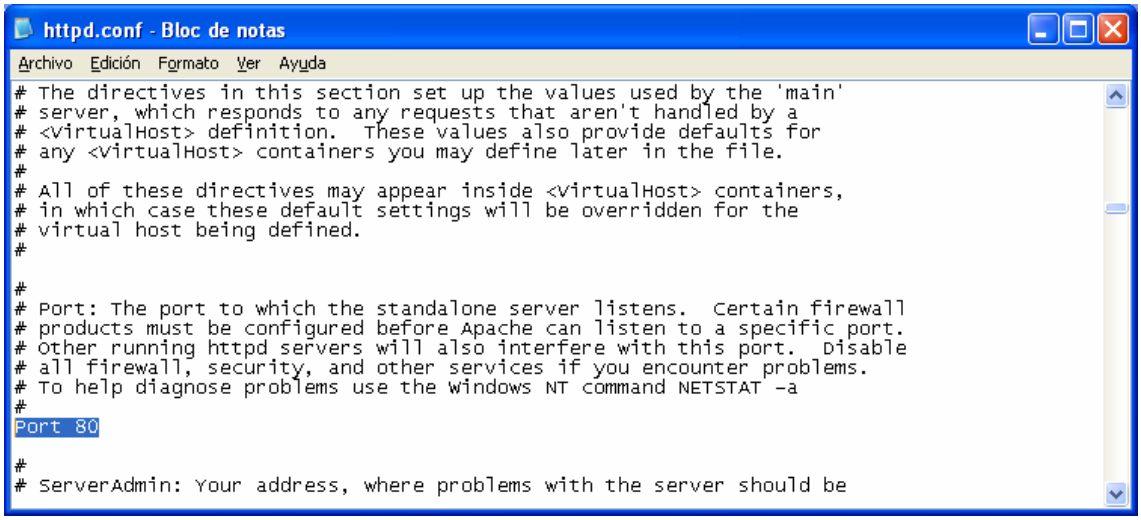

Figura 53: Configuración del puerto de Apache.

Para comprobar si el puerto deseado está disponible ejecute en línea de comandos la siguiente instrucción:

telnet localhost <puerto>

7.3.2 No Encuentra la Página Principal de MIGS

El servidor Apache está en funcionamiento pero no localiza la página principal de MIGS.

Compruebe que efectivamente la ruta es correcta. Recuerde que la URL es *case sensitive* y que si el puerto de Apache es distinto al 80 debe especificarlo. Ej puerto 8085:

http://localhost:8085/migs/principal/principal.html

7.3.3 Error de Conexión con la Base de Datos

He conseguido entrar a la página principal pero cuando quiero ver los objetos del museo da un error de conexión con la base de datos.

Asegúrese de que el servidor de la base de datos está en funcionamiento. Puede comprobarlo si en la barra de Windows aparece el semáforo de MySQL y está en verde.

También es posible que no esté correctamente configurada la conexión con la base de datos. Compruebe que las siguientes líneas son correctas en los archivos “CLASES/basedatos.php” y “CLASES/buscardatos.php” del directorio de la aplicación.

```
function BaseDatos() {
```

```
$this->conexion = mysql_pconnect("localhost","root","clave") //servidor, usuarioBD, claveBD
        or DIE("Error al conectarse a la base de datos");
mysql_select_db("migs"); //nombre de la Base de Datos
}
```

En servidores MAC es posible que no funcione si en el nombre del servidor escribimos el nombre de la máquina. Por ello especifique siempre el valor “localhost”.

7.3.4 Los Recursos no se Suben al Servidor

La navegación es correcta, pero al introducir objetos en el museo los recursos propios no se suben al servidor.

Se trata de un problema de permisos en el disco duro. Es necesario asignar permisos de escritura en todo el directorio de la aplicación así como en sus subcarpetas, especialmente en “recursos” y “privado”.

7.3.5 Error en las Descargas

Al intentar bajar o subir un objeto del museo da un error de PHP.

Es debido a la configuración de las extensiones de PHP. Cerciórese que la ruta que se especifico en el archivo php.ini es la correcta, que la carga de las librerías “php_domxml.dll” y “php_zip.dll” no esta comentada y que ambos archivos existen en el directorio “extensions” de PHP.

8 Bibliografía

1. IEEE/LTSC: http://ltsc.ieee.org/
2. IMS: www.imsglobal.org
3. ADL: www.adlnet.org
4. A. Fernández-Valmayor, M. Guinea, M. Jiménez, A. Navarro, A. Sarasa

Virtual objects an approach to building learning objects in archaelogy, in. M. Llamas-Nistal et al. (Eds) Computers and Education. Towards a Long-Life Learning Society 2003, pp.191- 202. Kluwer Academic Publisher. Dordrecht, Neederlans

5. LOM: http://ltsc.ieee.org/wg12/
6. L. Mortimer: http://www.learningcircuits.org/2002/apr2002/mortimer.html
7. Dublin Core: http://dublincore.org/
8. ARIADNE: http://www.ariadne.ac.uk/
9. Patricia Díaz Ayuso, Beatriz Díaz García, Concepción Sanz Pineda

Uso y Gestión de Objetos de Aprendizaje en la Web. Proyecto de Sistemas Informáticos de la Facultad de Informática de la Universidad Complutense de Madrid curso 2003/2004

10. Chasqui: http://macgalatea.sip.ucm.es/chasqui
11. Daniel Burgos, Colin Tattersall, Rob Koper

Utilización de estándares en el Aprendizaje Virtual. Funcionalidades didácticas de la especificación IMS Learning Design. Actas de la II jornada Campus Virtual UCM. Editorial Complutense (en prensa).

12. Patricia Díaz Ayuso, Beatriz Díaz García, Concepción Sanz Pineda.

El museo Virtual de Informática de García Santesmases. Editado en Campus Virtual Universidad Complutense de Madrid 2004.

13. Mercedes Guinea Bueno

El Proyecto Chasqui. Editado en Campus Virtual Universidad Complutense de Madrid 2004.

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto en la propia memoria, como el código, la documentación y/o el prototipo desarrollado para la asignatura de Sistemas Informáticos del curso 2004/2005 bajo el nombre “Creación, Gestión y Uso de Objetos de Aprendizaje en un Entorno Web”.

Los Autores:

Juan José Calvo Diaz de Neira

Alberto Espadero Guillermo

David Sánchez Llorente