
Tutorial para la creación de un módulo en Moodle

Índice

Tutorial para la creación de un módulo en Moodle.....	1
Introducción.....	2
1. Antes de empezar.....	2
1.1. Recomendaciones.....	2
1.2. Instalación de Moodle.....	2
2. Esquema de un módulo.....	2
2.1. El fichero o paquete de idioma.....	3
2.2. Permisos de un módulo.....	4
2.3 El módulo Tarea (assignment).....	5
2.3.1. Vista preliminar del código fuente.....	5
3. Creación de una actividad básica.....	8
3.1. Introducción.....	8
3.2. Desarrollo del módulo Ejemplo.....	8
3.2.1. Creación del formulario.....	8
3.2.2. Elaboración de las tablas.....	9
3.2.3. Creación del fichero de permisos.....	10
3.2.4. Edición del fichero lib.php.....	10
4. Instalación del módulo.....	11
4.1. Desinstalación de la actividad.....	11
Referencias.....	13
Anexos.....	14
Anexo 1. Código fuente Tarea (Assignment).....	14
Mod.html.....	14
Mysql.sql.....	16
Access.php.....	18
Anexo 2. Código fuente módulo "Ejemplo".....	18
Fichero mod.html.....	18
Fichero Mysql.sql.....	20
GNU Free Documentation License.....	21

Copyright (c) 2007 FRANCISCO JAVIER RODRÍGUEZ MARTÍN, jrm_500@yahoo.es
UNIVERSIDAD DE EXTREMADURA

Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled "GNU Free
Documentation License".

Introducción

Este tutorial es parte del trabajo realizado para el Proyecto Fin de Carrera titulado *Integración de un sistema de ejecución remota de programas en Moodle*. Este proyecto ha sido realizado en el curso 2006-2007 por el alumno *Francisco Javier Rodríguez Martín* en la Escuela Politécnica de Cáceres. El director del proyecto fue *Alberto Gómez Mancha*. Universidad de Extremadura.

Para este proyecto realicé el módulo *Remote Running* que compila y ejecuta programas de forma remota. Mediante este módulo, los alumnos en un curso de Moodle pueden compilar y ejecutar sus prácticas o problemas de programación de forma remota, sin tener instalado un compilador en sus ordenadores.

Con este tutorial se pretende simplificar la labor de creación de un nuevo módulo a los futuros desarrolladores de Moodle. Este documento nace con la idea de facilitar el camino, dar pautas de desarrollo a los programadores noveles y comentar aquellas incidencias que se han presentado en la creación de un nuevo módulo.

El código fuente propio de la plataforma que se muestra en este tutorial corresponde a la versión *1.6.3*.

1. Antes de empezar

1.1. Recomendaciones

Antes de iniciarse en la programación de cualquier aspecto de Moodle es necesario entender la plataforma, cómo funciona, para qué sirve cada opción, cuál es el papel del profesor, del programador o del administrador. Desde el sitio Web de Moodle se puede descargar manuales de la plataforma. [MANMOODLE]

Otra recomendación interesante y prácticamente obligatoria si se pretende que otros desarrolladores entiendan, mejoren o continúen el trabajo realizado es seguir el *manual de estilo de código* [MANCODIGO]. En este manual se dan una serie de reglas generales para el tratamiento de ficheros, funciones y variables de un módulo. También se comentan reglas de estilo de código, la estructura de la base de datos y las normas de seguridad en la creación de una actividad.

1.2. Instalación de Moodle

Antes de comenzar con el tema central de este tutorial se darán unas pautas básicas para instalar la plataforma. Todo el proceso de instalación según *Windows* o *Linux* está descrito en la web oficial de Moodle [MODINSTALL]. En este apartado se resumen los pasos de la instalación.

Moodle necesita los siguientes requisitos para ejecutarse:

- Un *servidor Web* como por ejemplo, *Apache*.
- Una instalación de *PHP*. Sirve cualquier versión a partir de la *4.3.0*.
- Una base de datos. *Mysql* es la más popular aunque *PostgreSQL* cuenta cada día con más seguidores.

Al paquete *Mysql*, *PHP* y *Apache* se le denomina *plataforma LAMP* y puede ser descargada e instalada desde la página de Moodle [MODLAMP]. También existe el paquete de software *Appserv*. Ésta es una herramienta *Opensource* para *Windows* que instala *Apache*, *Mysql* y *PHP* y configura las aplicaciones de manera automática. Existen multitud de tutoriales sencillos sobre esta herramienta para su instalación [TUTAPPSERV].

El siguiente paso es obtener la plataforma [MODDOWN] y descomprimir el archivo obtenido. Se debe colocar la carpeta completa en su servidor web (<http://suservidor.com/moodle>) y a continuación ejecutar el *script* de instalación usando un navegador común. Para acceder a este *script* simplemente habrá que redirigir el navegador a la dirección <http://suservidor.com/moodle/install.php>. El proceso de instalación es automático, requiriéndose la intervención del usuario en contadas ocasiones.

Una vez realizado el proceso de instalación se podrá ejecutar Moodle desde la dirección <http://susservidor.com/moodle>, asegurándose el usuario de que *Apache* y *Mysql* están ejecutándose.

Existe una lista de preguntas frecuentes sobre la instalación de Moodle [MODFAQ] e información para la realización de actualizaciones del software de la plataforma en la Web [MODUPDATE].

2. Esquema de un módulo

Previo al desarrollo de una actividad es necesario conocer el esquema de un módulo. Los módulos se almacenan en la carpeta *moodle/mod*, cada uno en un directorio, siendo la estructura general de archivos y directorios como se explica a continuación [MODACT]:

- *mod.html*: formulario para configurar o actualizar las opciones del módulo.
- *version.php*: para informar de la versión del módulo y código de actualización.
- *lang/*: directorio para almacenar el fichero de idioma del módulo. Esto será explicado en el apartado 3.1.
- *db/*: esto es un directorio. Aquí se almacenarán los ficheros con las tablas de las bases de datos necesarias para la actividad. Por ejemplo, si se utiliza *mysql*, irán los ficheros *mysql.sql* y *mysql.php*. El fichero *.sql* ha de tener al menos una tabla con las características mínimas del módulo. Esto será explicado en el apartado 3.2.2.
- *db/access.php*: fichero de permisos del módulo. Los permisos no son obligatorios pero sí muy recomendables para desarrollar módulos eficientes. Comentado en el apartado 2.2.
- *index.php*: este fichero sirve para mostrar todas las instancias de una actividad, es decir, una lista con todas las instancias del mismo módulo.
- *view.php*: esta es la página de inicio de la actividad.
- *lib.php*: librería de funciones del módulo. En este fichero se implementarán todas las funciones y procedimientos del módulo. Si el módulo se llama *inicio*, entonces las funciones mínimas y obligatorias que ha de tener la actividad tienen que ser de la forma:
 - *inicio_add_instance()*: código para añadir una nueva instancia.
 - *inicio_update_instance()*: función para actualizar una instancia existente.
 - *inicio_delete_instance()*: código para borrar una instancia.
 - *inicio_user_outline()*: da un resumen concreto de la actividad de un usuario.
 - *inicio_user_complete()*: devuelve un informe más detallado de un usuario.
- Todas las funciones, procedimientos y constantes, creados en *lib.php*, tienen que comenzar con el nombre del módulo.
- El archivo *config.html* es opcional, permitiendo a los administradores configurar opciones generales del módulo.
- El módulo debe tener archivos de idioma que contenga las cadenas para ese módulo. Deberá ser programado en inglés y traducido al idioma de los usuarios finales que utilicen la actividad. Esto está explicado en el siguiente apartado.

2.1. El fichero o paquete de idioma

Moodle es una plataforma internacional. Cada paquete de idioma que se quiera definir ha de crearse como un *archivo de idioma* de forma que cada cadena que forme parte de la interfaz, se extraiga de un conjunto de archivos de idioma. Toda la información respecto a la configuración de idioma, traducciones y creación de paquetes de idiomas se puede obtener del sitio de Moodle [MODTRAD], resumido en estas líneas:

- *Estructura del paquete de idioma*: los paquetes de idiomas se encuentran en la carpeta *moodledata/lang* con nombres que indican que son paquetes de idioma *utf-8*. Para el caso del español internacional el nombre de la carpeta es *es_utf8*. El idioma por defecto es el inglés *en_utf8* y se encuentra en una carpeta diferente (*moodle/lang*).
- *Creación del paquete de idioma*: para desarrollar un fichero de idioma para el módulo que se creará y teniendo en cuenta el caso del español internacional (*es_utf8*), hay que crear un fichero con el mismo nombre que el módulo en la carpeta *lang/es_es_utf8* y con extensión *.php*. Por ejemplo, si el módulo se denomina *inicio*, el fichero se llamará *inicio.php* y estará en la carpeta *moodle/mod/inicio/lang/es_es_utf8*, compuesto de líneas de texto cortas denominadas *cadenas* (*string*) como se muestra en la figura 1.

```
//moodle
$string['modulename'] = 'Inicio';
$string['modulenameplural'] = 'Inicios';

//Del propio módulo
$string['Name'] = 'Nombre para la actividad Inicio';
$string['name'] = 'Nombre para la actividad Inicio';
$string['description'] = 'Descripción';
$string['grade'] = 'Calificación de Inicio';
...
```

Figura 1. Ejemplo de código para el fichero de idioma

Las primeras entradas hacen referencia a variables que utiliza Moodle (*modulename*, *modulenameplural*). Las variables *name*, *description* y *grade* son propias del módulo *ejemplo* y se utilizan en el formulario de inserción de datos. Se utilizan de la siguiente manera:

```
print_string("name","inicio")
get_string("name", "inicio")
```

- *Errores*: si una cadena no existe en un idioma en particular, se mostrará por defecto su equivalente en inglés. Esto puede producir en determinadas ocasiones la mezcla de diferentes lenguajes (normalmente el inglés más el definido por el usuario). Este fallo se corrige con la inserción de la cadena que produce el fallo en el fichero de idioma correspondiente.

Antes de comenzar con el desarrollo de un módulo se va a analizar el módulo *tarea* en el apartado siguiente. Será un análisis sobre el código fuente, para entender los ficheros más importantes de un módulo antes de comenzar a crear uno nuevo. También hay que dominar ciertos aspectos:

- Comprender perfectamente el esquema de un módulo para saber qué hace cada fichero y donde se programará cada apartado (formularios, tablas, funciones, etc).
- Dominar el lenguaje *PHP* y tener conocimiento de *html*. *PHP* es el lenguaje utilizado para programar la actividad y *html* será utilizado en algunas fases de desarrollo, como, por ejemplo, a la hora de crear el formulario de opciones.
- Tener conocimientos de *sql* para la creación de las tablas de la base de datos.

2.2. Permisos de un módulo

Los permisos permiten establecer las diferentes vistas que tendrán los distintos tipos de usuarios para un módulo. Los tipos de usuarios son los *roles* en Moodle [MOODROL]. Ejemplos de roles son administrador, profesor, estudiante, etc.

Asignando permisos a la actividad se puede restringir el acceso a cierta información, configurar distintos tipos de vistas para los diferentes roles, establecer capacidades para los administradores, etc.

Para establecer permisos en la actividad, se creará un fichero denominado *access.php*, en la carpeta *db*. Si el módulo se denomina *inicio* y queremos establecer que usuario (*rol*) tendrá acceso a la vista del mismo, y que usuario puede calificar el módulo, el fichero quedaría de la siguiente manera:

```
<?php

$mod_inicio_capabilities = array(

    'mod/inicio:view' => array(

        'capytype' => 'read',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'guest' => CAP_ALLOW,
            'student' => CAP_ALLOW,
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'admin' => CAP_ALLOW
        )
    ),

    'mod/inicio:grade' => array(

        'capytype' => 'write',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'admin' => CAP_ALLOW
        )
    )
);

?>
```

CONTEXT_MODULE establece el contexto en el que se aplicará el permiso, en este caso, al módulo, y *CAP_ALLOW* un tipo de habilidad que se asigna a los roles.

La forma de utilizar estos permisos es muy sencilla. Por ejemplo, en el fichero *view.php*, si solo se quiere permitir la vista a los roles definidos, el código sería el siguiente:

```
//Anterior código de view.php
```

```
$context = get_context_instance(CONTEXT_MODULE, $cm->id);
require_capability('mod/inicio:view', $context);

//Resto de código de view.php
```

Los permisos se crean cuando se instala el módulo. Para comprobar que Moodle ha incluido los permisos adecuadamente, habrá que dirigirse, como administrador, a USERS -> PERMISSIONS -> DEFINE ROLES y pinchar sobre el rol a comprobar. Más información sobre permisos, roles, contextos y habilidades en la web de Moodle [MOODROLD].

2.3 El módulo Tarea (assignment)

2.3.1. Vista preliminar del código fuente

Analizando el código fuente del módulo *tarea* (Anexo 1) se descubren aspectos interesantes que podrán aplicarse al programar una actividad. Los ficheros que se estudiarán en este apartado son los que corresponden al formulario de configuración del módulo (*mod.html*), la librería de funciones del módulo (*lib.php*), los ficheros de la base de datos (*mysql.sql*, *mysql.php*) y el fichero de permisos (*access.php*).

Mod.html

El primer archivo de código fuente que se va a estudiar es el formulario *mod.html*. Los formularios se lanzan cuando se configura el módulo y son muy importantes porque establecen las opciones de la actividad.

Al principio del fichero se observa cómo se crean e inicializan las variables necesarias del formulario con sentencias condicionales *if*. Las variables inicializadas son aquellas que formarán parte del formulario, como por ejemplo el nombre de la tarea (*name*), una descripción del contenido o de la actividad (*description*), calificación máxima que tendrá (*grade*), etc. Este código será *PHP*. Un poco más abajo y ya en código *html* se creará el formulario con las distintas preguntas y opciones de respuestas.

```
<?php
    require_once("$CFG->dirroot/mod/assignment/lib.php");

    if (empty($form->name)) {
        $form->name = "";
    }
    if (empty($form->description)) {
        $form->description = "";
    }
    if (empty($form->format)) {
        $form->format = "";
    }
    if (empty($form->assignmenttype)) {
        $form->assignmenttype = "uploadsingle";
    }
    if (!isset($form->grade)) {
        $form->grade = 100;
    }
    if (!isset($form->timedue)) {
        $form->timedue = time()+7*24*3600;
    }
    if (!isset($form->timeavailable)) {
        $form->timeavailable = time();
    }
    if (!isset($form->preventlate)) {
        $form->preventlate = 0;
    }
    if (empty($form->maxbytes)) {
        $form->maxbytes = $CFG->assignment_maxbytes;
    }
    ...
```

Figura 2. Creación e inicialización de variables del formulario tarea en PHP

Como se puede comprobar también hay código *PHP* insertado para, por ejemplo, llamar a un editor *html*¹ ya definido en Moodle o llamar a variables que contienen el nombre de las respectivas opciones del formulario. Y poco más que comentar de este fichero, es un simple formulario con la única dificultad de entender su contenido.

Lib.php

Otro fichero muy importante es *lib.php*. En este archivo se programará generalmente, todo lo relacionado con librería de funciones y constantes del módulo, es decir, todas las funciones, procedimientos y constantes que se necesiten estarán definidas en este fichero. También se declararán aquí los objetos que se utilicen:

```
class assignment_base {  
  
    var $cm;  
    var $course;  
    var $assignment;  
    var $strassignment;  
    var $strassignments;  
    var $strsubmissions;  
    var $strlastmodified;  
    var $navigation;  
    var $pagetitle;  
    var $currentgroup;  
    var $usehtmleditor;  
    var $defaultformat;  
  
    ...  
}
```

Figura 3. Ejemplo de código de *lib.php*

MySQL.sql y MySQL.php

Continuando con esta introducción al módulo tarea se comentará otros dos ficheros importantísimos a la hora de realizar un nuevo módulo, los ficheros para crear la base de datos del módulo.

Los datos en Moodle se almacenan en tablas creadas por una determinada base de datos. No se entrará en detalles ya que más adelante esto será explicado. Los ficheros importantes son *mysql.php* y *mysql.sql*, que se ubican en *mod/assignment/db*. En el fichero con extensión *sql* se escribirá el código de las tablas que se crearán cuando se instale el módulo:

¹ Un editor *html* es una interfaz para generar código html sin la necesidad de conocer el lenguaje. Da formato al texto y ofrece otras características [MODEEDITOR].

```

CREATE TABLE `prefix_assignment` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `course` int(10) unsigned NOT NULL default '0',
  `name` varchar(255) NOT NULL default '',
  `description` text NOT NULL default '',
  `format` tinyint(4) unsigned NOT NULL default '0',
  `assignmenttype` varchar(50) NOT NULL default '',
  `resubmit` tinyint(2) unsigned NOT NULL default '0',
  `preventlate` tinyint(2) unsigned NOT NULL default '0',
  `emailteachers` tinyint(2) unsigned NOT NULL default '0',
  `var1` int(10) default '0',
  `var2` int(10) default '0',
  `var3` int(10) default '0',
  `var4` int(10) default '0',
  `var5` int(10) default '0',
  `maxbytes` int(10) unsigned NOT NULL default '100000',
  `timedue` int(10) unsigned NOT NULL default '0',
  `timeavailable` int(10) unsigned NOT NULL default '0',
  `grade` int(10) NOT NULL default '0',
  `timemodified` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `course` (`course`)
) COMMENT='Defines assignments';

```

Figura 4. Ejemplo de creación de tablas del módulo tarea

El fichero con extensión *php* sirve para actualizar la base de datos para versiones anteriores. Por ejemplo, se puede añadir código específico para actualizar la base de datos si el módulo se va a instalar en versiones anteriores de Moodle. Ejemplo:

```

if ($oldversion < 2002080500) {

    execute_sql("
    CREATE TABLE `assignment` (
      `id` int(10) unsigned NOT NULL auto_increment,
      `course` int(10) unsigned NOT NULL default '0',
      `name` varchar(255) NOT NULL default '',
      `description` text NOT NULL,
      `type` int(10) unsigned NOT NULL default '1',
      `maxbytes` int(10) unsigned NOT NULL default '100000',
      `timedue` int(10) unsigned NOT NULL default '0',
      `grade` int(10) NOT NULL default '0',
      `timemodified` int(10) unsigned NOT NULL default '0',
      PRIMARY KEY (`id`)
    ) COMMENT='Defines assignments'
    ");

```

Figura 5. Código del fichero mysql.php

Access.php

Para finalizar con el modulo tarea, se mostrará el código de los permisos definidos para esta actividad:

```
$mod_assignment_capabilities = array(

    'mod/assignment:view' => array(

        'capytype' => 'read',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'guest' => CAP_ALLOW,
            'student' => CAP_ALLOW,
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'admin' => CAP_ALLOW
        )
    ),

    'mod/assignment:submit' => array(

        'capytype' => 'write',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'student' => CAP_ALLOW
        )
    ),

    'mod/assignment:grade' => array(

        'capytype' => 'write',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'admin' => CAP_ALLOW
        )
    )
);
```

Figura 6. Código del fichero access.php

3. Creación de una actividad básica

3.1. Introducción

En este apartado se tratará el desarrollo y los problemas que se pueden obtener en la realización de un módulo sencillo titulado *Ejemplo*. Este módulo no hace nada, simplemente pide una serie de datos y los almacena en tablas oportunas.

El sitio Web de Moodle dispone de una plantilla para realizar módulos. Esta plantilla se denomina *Newmodule* y es un fichero comprimido que se puede conseguir desde la Web de Moodle [NEWMOD].

La estructura de directorios y ficheros ya se ha tratado en puntos anteriores. Brevemente y para comenzar con este módulo recordar que lo más interesante es lo siguiente:

- *Mod.html*: fichero de formulario, se utiliza para crear el formulario de introducción de datos al crear una instancia del módulo.
- *db/mysql.sql*: fichero de creación de tablas con código sql.
- *db/access.php*: fichero para los permisos y roles.
- *lib.php*: este fichero sirve para crear las funciones que utiliza el módulo, funciona como librería de funciones y procedimientos.

Los pasos básicos para realizar un módulo son:

- Desarrollo del formulario para obtener los datos de configuración del módulo.
- Creación de las tablas en el directorio *db/mysql.sql*.
- Edición de los permisos del módulo.
- Edición de la librería de funciones del archivo *lib.php*.

¿Por qué seguir estas etapas? Porque cuando se desarrolla un módulo lo primero que hay que conocer es qué opciones tendrá (nombre del módulo, descripción, calificación, fecha tope de entrega, etc.), cómo se almacenarán esos datos (en las tablas de la base de datos), editar los permisos para los roles si son necesarios y cómo crear una instancia cuando la actividad esté configurada.

3.2. Desarrollo del módulo Ejemplo

Una vez que se tiene claro las opciones con las que contará el módulo, se comenzará a editar el código fuente. En este apartado se va a desarrollar los pasos básicos para realizar un módulo nuevo, comentando al final ciertos errores que se pueden producir en la creación de la actividad.

3.2.1. Creación del formulario

Para crear este módulo sencillo basta con varias opciones de configuración como por ejemplo un nombre, una breve descripción (mediante un editor *html*) y una calificación. Posteriormente se tendrá que editar el código del fichero *mod.html* para crear el formulario, pero al comienzo del mismo hay que añadir las variables oportunas para recoger las opciones deseadas. Estas variables pueden ser las siguientes: *name*, *description* y *grade*², declarándolas al comienzo del fichero de la siguiente forma:

```
if (!isset($form->name)) {
    $form->name = '';
}
if (!isset($form->description)) {
    $form->description = "";
} if (!isset($form->grade)) {
    $form->grade = 100; //100 es el valor predeterminado que tendrá.
}
```

Figura 7. Declaración de variables para el formulario del módulo Ejemplo

² Una de las normas de estilo de código de Moodle es programar en inglés para que toda la comunidad de desarrolladores pueda entender el código realizado.

Una vez realizado lo anterior ya se puede modificar el código del formulario que viene predefinido en el fichero. Solo hay que añadir líneas del estilo siguiente:

```
...
<td align="right"><b><?php print_string("name", "ejemplo") ?></b></td>
<td>
  <input type="text" name="name" size=60 value="<?php p($form->name) ?>">
</td>
</tr>
<tr valign="top">
  <td align="right"><b><?php print_string("description", "ejemplo") ?></b>
  </td>
  ...
```

Con la función `print_string` ("name", "ejemplo"), situada en el código del formulario anterior, se utiliza el fichero de idioma previamente creado, es decir, el sistema devuelve la cadena `name` del fichero de idioma `ejemplo`. Una vez insertado el módulo en Moodle, proceso que se explica en apartados posteriores de este tutorial, y configurado el fichero de idioma como se indica en el apartado 2.1, se obtiene un formulario como el de la *figura 8*.

Figura 8. Formulario del módulo ejemplo

3.2.2. Elaboración de las tablas

El siguiente paso en el desarrollo de la actividad es elaborar las tablas que permitirán almacenar los datos del módulo. Para crear las tablas hay que tener muy claro que es lo que se quiere conseguir con el módulo ya que las tablas que se necesiten dependerán de lo que se quiera implementar.

Como se indica en el manual de estilo de código [MANCODIGO] las tablas tienen que tener obligatoriamente un campo `id` (int 10) como clave primaria. Todo módulo necesita una tabla principal para contener instancias de cada módulo y debe tener el mismo nombre que el módulo y contener, como mínimo, los siguientes campos:

- `id`, descrito anteriormente.
- `course`, el identificador del curso al que pertenece la instancia de la actividad.
- `name`, el nombre de la instancia.

Otra norma interesante es que la mayoría de las tablas han de tener un campo denominado *timemodified* que será actualizado con la fecha actual del sistema, en el momento en que se actualice o cree la actividad.

Para este módulo basta con crear una tabla básica que recoja las opciones del formulario. También se añade el campo *timemodified* para que guarde el instante en que se modifica el módulo. Recordemos que las tablas se crean en el fichero *mysql.sql*, dentro del directorio *db*. El código puede ser algo parecido a esto:

```
CREATE TABLE `prefix_ejemplo` (
  `id` int(10) unsigned NOT NULL auto_increment, //ESTE ATRIBUTO NUNCA DEBE FALTAR
  `course` int(10) unsigned NOT NULL default '0',
  `name` varchar(255) NOT NULL default "",
  `description` text NOT NULL,
  `grade` int(10) NOT NULL default '0',
  `timemodified` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`)
) COMMENT='Define Ejemplo';
```

3.2.3. Creación del fichero de permisos

Este módulo de ejemplo es tan sencillo que no requiere ningún tipo de permisos para los roles, pero en el caso de que se requirieran, éste sería el momento de crear el fichero *access.php* en la carpeta *mod/ejemplo/db*. Si se necesitara que el profesor fuera el único con derecho a evaluar la actividad, el código sería el siguiente:

```
<?php

$mod_ejemplo_capabilities = array(

'mod/assignment:grade' => array(

    'captype' => 'write',
    'contextlevel' => CONTEXT_MODULE,
    'legacy' => array(
        'teacher' => CAP_ALLOW,
        'editingteacher' => CAP_ALLOW,
        'admin' => CAP_ALLOW
    )
)
);

?>
```

No es necesario negar permisos a los demás roles, simplemente con darle el permiso *CAP_ALLOW* al profesor (*teacher*) ya se suponen prohibidos los permisos para los demás.

3.2.4. Edición del fichero lib.php

Y una vez realizado el paso anterior, solo queda definir la función que añada una instancia del módulo a la plataforma. Esta función que hay que diseñar se encuentra en */mod/lib.php* y deberá comenzar con el nombre del módulo. La función ya es nombrada anteriormente y es *****_add_instance()* que en este caso quedaría de la siguiente manera:

```
function ejemplo_add_instance($ejemplo) {

    $ejemplo->timemodified = time();
    if ($mod->course) {
        if (! $course = get_record("course", "id", $mod->course)) {
            error("Course is misconfigured");
        }
    }

    # May have to add extra stuff in here #
    return insert_record("ejemplo",$ejemplo);
}
```

Y con estos sencillos pasos ya se tendría un primer módulo de prueba diseñado. A partir de este punto, habría que editar el código del fichero *view.php* para que el módulo tuviera funcionalidad. Simplemente comentar ciertos errores que hay que evitar desarrollando un módulo:

- Una vez que se termina el formulario, en el mismo fichero *mod.html* hay que insertar o añadir las siguientes líneas de código:

```
<input type="hidden" name=course      value="<?php p($form->course) ?>" />
<input type="hidden" name=module      value="<?php p($form->module) ?>" />
<input type="hidden" name=modulename  value="<?php p($form->modulename) ?>" />
<input type="hidden" name=instance    value="<?php p($form->instance) ?>" />
<input type="hidden" name=sesskey     value="<?php p($form->sesskey) ?>" />
<input type="hidden" name=mode        value="<?php p($form->mode) ?>" />
<input type="hidden" name=section     value="<?php p($form->section) ?>" />
```

Si no son añadidas, se producirán fallos de seguridad y avisos de que faltan ciertos parámetros.

- En la creación de tablas, el atributo *id* es obligatorio y hay que insertarlo exactamente como se indica en líneas anteriores.

3.2.5. Botones de ayuda

Un asunto importante a la hora de desarrollar un módulo es introducir botones de ayuda para explicar ciertos aspectos a los usuarios. Por ejemplo, para explicar alguna opción, botones de la interfaz que no esté clara su utilidad y en general, en aquellos sitios que se crea recomendable.

Para crear estos botones de ayuda, es necesario crear archivos *html* en la carpeta de idioma del módulo. Siguiendo con el módulo “Ejemplo”, y el caso del idioma español (es_es_utf8), se crean en *ejemplo/lang/es_es_utf8/help/ejemplo* y es necesario llamar a la siguiente función en *PHP*:

```
helpbutton("description", get_string("description", "ejemplo"), "ejemplo");
```

El primer parámetro hace referencia al nombre del fichero *html*, es decir, a *description.html*. El segundo parámetro será una cadena de texto que aparecerá al situar el cursor sobre el botón de ayuda y el tercer parámetro el nombre del módulo. Las figuras 9 y 10 muestran el botón de ayuda al lado de la opción *descripción* y la ventana que se muestra al pinchar sobre dicho botón.

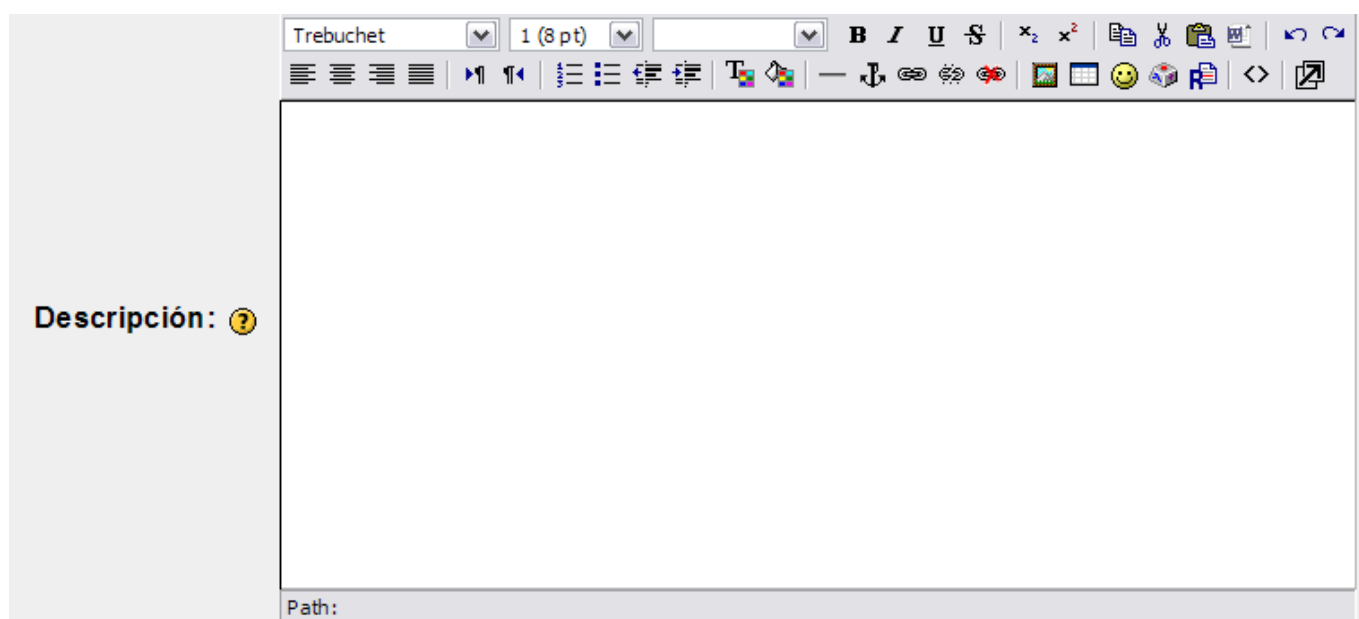


Figura 9. Botón de ayuda para opción descripción



Figura 10. Ventana de ayuda para el botón descripción

4. Instalación del módulo

Para integrar un módulo en Moodle hay que seguir unos pasos realmente sencillos. Una vez que el módulo está creado y programado correctamente, habrá que situarlo dentro de la carpeta `...moodle/mod`. En esta carpeta es donde están situados todos los módulos del programa.

El siguiente paso es entrar en el sistema como *administrador* y dirigir el navegador hacia la página `http://moodlesite/moodle/admin/index.php`. Una vez realizado este paso se crearán las tablas de la base de datos automáticamente y se podrá usar el módulo. *Moodlesite* hace referencia al servidor donde esté alojada la plataforma.

4.1. Desinstalación de la actividad

Para eliminar un módulo instalado hay que acceder a la plataforma como administrador y situarse en la página de *actividades* desde la ruta *Administración* → *Módulos* → *Actividades*, pinchando en el enlace borrar del módulo que se desee desinstalar, como se puede comprobar en la figura siguiente:

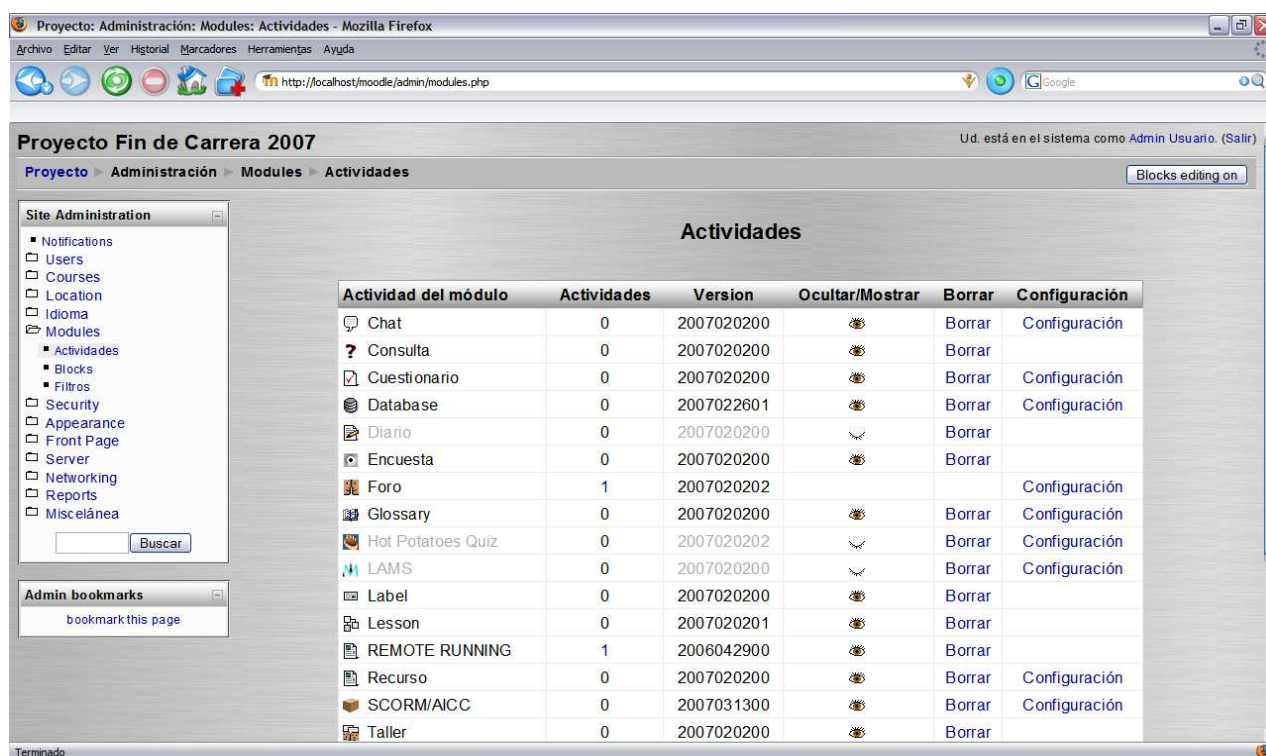


Figura 11. Configuración de los módulos

Para una correcta eliminación hay que borrar las tablas creadas por los módulos. Este proceso suele ser automático, pero se ha comprobado que a veces la plataforma no elimina las tablas adecuadamente y hay que realizar el proceso de forma manual. El siguiente ejemplo es para las bases de datos *Mysql*, con el módulo *Ejemplo*:

1. Iniciar un terminal.
2. `Mysql -u root -p`
3. `connect moodle;`
4. `drop table mdl_ejemplo;`
5. `quit`

Mediante los comandos anteriores, se desinstalaría el módulo correctamente. Hay que tener en cuenta que *moodle* es el nombre de la base de datos y que las tablas tienen el prefijo *mdl_*.

Referencias

[MANCODIGO] Moodle: “Manual de estilo de código” ([http://docs.moodle.org/es/Manual de Estilo de C%C3%B3digo](http://docs.moodle.org/es/Manual_de_Estilo_de_C%C3%B3digo)), Marzo 2007.

[MANMOODLE] Manuales de Moodle ([http://docs.moodle.org/es/Manuales de Moodle](http://docs.moodle.org/es/Manuales_de_Moodle)), Marzo 2007.

[MODACT] Moodle: “Módulo de actividades”, ([http://docs.moodle.org/es/M%C3%B3dulos de actividades %28desarrollador%29](http://docs.moodle.org/es/M%C3%B3dulos_de_actividades_%28desarrollador%29)), Marzo 2007.

[MODDOWN] Moodle: “Download” (<http://download.moodle.org/>), Marzo 2007.

[MODEEDITOR] Moodle: “Editor html” ([http://docs.moodle.org/es/Editor HTML](http://docs.moodle.org/es/Editor_HTML)), Marzo 2007.

[MODFAQ] Moodle: “FAQ Instalación” ([http://docs.moodle.org/es/FAQ Instalaci%C3%B3n](http://docs.moodle.org/es/FAQ_Instalaci%C3%B3n)) Marzo 2007.

[MODINSTALL] Moodle: “Instalación de Moodle” ([http://docs.moodle.org/es/Instalaci%C3%B3n de moodle](http://docs.moodle.org/es/Instalaci%C3%B3n_de_moodle)), Marzo 2007.

[MODLAMP] Moodle: “Instalación AMP” ([http://docs.moodle.org/es/Instalaci%C3%B3n AMP](http://docs.moodle.org/es/Instalaci%C3%B3n_AMP)), Marzo 2007.

[MOODROL] Moodle: “Roles” (<http://docs.moodle.org/es/Roles>), Junio 2007.

[MOODROLD] Moodle: “Desarrollador: Roles” (<http://docs.moodle.org/es/Desarrollador:Roles>), Junio 2007.

[MODTRAD] Moodle: “Traducciones” (<http://docs.moodle.org/es/Traducciones>), Marzo 2007.

[MODUPDATE] Moodle: “Actualización de Moodle” ([http://docs.moodle.org/es/Actualizaci%C3%B3n de moodle](http://docs.moodle.org/es/Actualizaci%C3%B3n_de_moodle)), Marzo 2007.

[NEWMOD] Moodle: “Plantilla para nuevo módulo” (<http://download.moodle.org/download.php/modules/NEWMODULE.zip>), Marzo 2007.

[TUTAPPSERV] The Stromboli Project: “Tutorial Appserv” (<http://www.thestromboliproject.com/tutoriales/extras/appserv-apache+-mysql-2.htm>), Marzo 2007.

Anexos

Anexo 1. Código fuente Tarea (Assignment)

Mod.html

```

<?php
    require_once("$CFG->dirroot/mod/assignment/lib.php");

    if (empty($form->name)) {
        $form->name = "";
    }
    if (empty($form->description)) {
        $form->description = "";
    }
    if (empty($form->format)) {
        $form->format = "";
    }
    if (empty($form->assignmenttype)) {
        $form->assignmenttype = "uploadsingle";
    }
    if (!isset($form->grade)) {
        $form->grade = 100;
    }
    if (!isset($form->timedue)) {
        $form->timedue = time()+7*24*3600;
    }
    if (!isset($form->timeavailable)) {
        $form->timeavailable = time();
    }
    if (!isset($form->preventlate)) {
        $form->preventlate = 0;
    }
    if (empty($form->maxbytes)) {
        $form->maxbytes = $CFG->assignment_maxbytes;
    }

?>
<script type="text/javascript" language="javascript">
    var dueitems = ['dueday','duemonth','dueyear','duehour', 'dueminute', 'preventlate'];
    var availableitems = ['availableday','availablemonth','availableyear','availablehour', 'availableminute'];
</script>

<form name="form" method="post" action="../mod/assignment/details.php">
<table cellpadding="5">
<tr valign="top">
    <td align="right"><b><?php print_string("assignmentname", "assignment") ?>:</b></td>
    <td>
        <input type="text" name="name" size=60 value="<?php p($form->name) ?>">
    </td>
</tr>
<tr valign="top">
    <td align="right"><b><?php print_string("description", "assignment") ?>:</b>
    <br /><br />
    <?php
        helpbutton("writing", get_string("helpwriting"), "moodle", true, true);
        echo "<br />";
        helpbutton("questions", get_string("helpquestions"), "moodle", true, true);
        echo "<br />";
        if ($usehtmleditor) {
            helpbutton("richtext", get_string("helprichtext"), "moodle", true, true);

```

```

    } else {
        emoticonhelpbutton("form", "description");
    }
?>
</td>
<td>
<?php
print_textarea($usehtmleditor, 20, 60, 680, 400, "description", $form->description);

if ($usehtmleditor) {
    echo '<input type="hidden" name="format" value="'.FORMAT_HTML.'" />';
} else {
    echo '<div align="right">';
    helpbutton("textformat", get_string("formattexttype"));
    print_string("formattexttype");
    echo ':&nbsp;';
    if (!$form->format) {
        $form->format = $defaultformat;
    }
    choose_from_menu(format_text_menu(), "format", $form->format, "");
    echo '</div>';
}
?>
</td>
</tr>
<tr valign=top>
<td align="right"><b><?php print_string("grade") ?>:</b></td>
<td>
<?php
print_grade_menu($course->id, "grade", "$form->grade");
?>
</td>
</tr>

<tr valign=top>
<td align="right"><b><?php print_string("availabledate", "assignment") ?>:</b></td>
<td>
<input name="availableenable" type="checkbox" value="1" alt="<?php print_string('availabledate', 'assignment') ?>"
onclick="return lockoptions('form', 'availableenable', availableitems)" <?php if ($form->timeavailable) echo
'checked="checked"' ?> />
<?php
print_date_selector("availableday", "availablemonth", "availableyear", $form->timeavailable);
echo "&nbsp;&nbsp;&nbsp;";
print_time_selector("availablehour", "availableminute", $form->timeavailable);
?>
<input type="hidden" name="havailableday" value="0" />
<input type="hidden" name="havailablemonth" value="0" />
<input type="hidden" name="havailableyear" value="0" />
<input type="hidden" name="havailablehour" value="0" />
<input type="hidden" name="havailableminute" value="0" />
</td>
</tr>

<tr valign=top>
<td align="right"><b><?php print_string("duedate", "assignment") ?>:</b></td>
<td>
<input name="dueenable" type="checkbox" value="1" alt="<?php print_string('duedate', 'assignment') ?>"
onclick="return lockoptions('form', 'dueenable', dueitems)" <?php if ($form->timedue) echo 'checked="checked"' ?> />
<?php
print_date_selector("dueday", "duemonth", "duyear", $form->timedue);
echo "&nbsp;&nbsp;&nbsp;";
print_time_selector("duehour", "dueminute", $form->timedue);
?>
<input type="hidden" name="hdueday" value="0" />
<input type="hidden" name="hduemonth" value="0" />

```

```

        <input type="hidden" name="hdueyear" value="0" />
        <input type="hidden" name="hduehour" value="0" />
        <input type="hidden" name="hdueminute" value="0" />
    </td>
</tr>

<tr valign="top">
    <td>&nbsp;</td>
    <td><?php print_string('preventlate', 'assignment') ?>:
    <?php
        $options[0] = get_string('no'); $options[1] = get_string('yes');
        choose_from_menu($options, 'preventlate', $form->preventlate, "");
    ?>
    <input type="hidden" name="hpreventlate" value="0" />
</td>
</tr>

<tr valign="top">
    <td colspan="2">&nbsp;</td>
</tr>

<tr valign="top">
    <td align="right"><b><?php print_string("assignmenttype", "assignment") ?></b></td>
    <td>
    <?php
        $ASSIGNMENT_TYPES = assignment_types();
        choose_from_menu($ASSIGNMENT_TYPES, 'assignmenttype', $form->assignmenttype, "");
        helpbutton('types', get_string('assignmenttype', 'assignment'), 'assignment');
    ?>
    </td>
</tr>

<?php print_standard_coursemodule_settings($form); ?>

</table>
<br />
<center>
<?php
    // Add hidden fields for all the ones not defined on this page
    $assvars = array('course','coursemodule','section','module','modulename','instance','mode',
        'resubmit','emailteachers','maxbytes', 'var1','var2','var3','var4','var5');

    foreach ($assvars as $assname) {
        if (isset($form->$assname)) {
            echo '<input type="hidden" name=".'$assname.'" value="'.s($form->$assname).'" />.'"\n";
        }
    }
?>
<input type="submit" value="<?php print_string('next') ?> &raquo;" />
<input type="submit" name="cancel" value="<?php print_string('cancel') ?>" />
</center>
</form>

<script type="text/javascript">
<?php
    if (!$form->timedue) echo "lockoptions('form','dueenable', dueitems);";
    if (!$form->timeavailable) echo "lockoptions('form','availableenable', availableitems);";
?>
</script>

```

Mysql.sql

```

#
# Table structure for table `assignment`
#

CREATE TABLE `prefix_assignment` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `course` int(10) unsigned NOT NULL default '0',
  `name` varchar(255) NOT NULL default "",
  `description` text NOT NULL default "",
  `format` tinyint(4) unsigned NOT NULL default '0',
  `assignmenttype` varchar(50) NOT NULL default "",
  `resubmit` tinyint(2) unsigned NOT NULL default '0',
  `preventlate` tinyint(2) unsigned NOT NULL default '0',
  `emailteachers` tinyint(2) unsigned NOT NULL default '0',
  `var1` int(10) default '0',
  `var2` int(10) default '0',
  `var3` int(10) default '0',
  `var4` int(10) default '0',
  `var5` int(10) default '0',
  `maxbytes` int(10) unsigned NOT NULL default '100000',
  `timedue` int(10) unsigned NOT NULL default '0',
  `timeavailable` int(10) unsigned NOT NULL default '0',
  `grade` int(10) NOT NULL default '0',
  `timemodified` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `course` (`course`)
) COMMENT='Defines assignments';
# -----

#
# Table structure for table `assignment_submissions`
#

CREATE TABLE `prefix_assignment_submissions` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `assignment` int(10) unsigned NOT NULL default '0',
  `userid` int(10) unsigned NOT NULL default '0',
  `timecreated` int(10) unsigned NOT NULL default '0',
  `timemodified` int(10) unsigned NOT NULL default '0',
  `numfiles` int(10) unsigned NOT NULL default '0',
  `data1` text NOT NULL default "",
  `data2` text NOT NULL default "",
  `grade` int(11) NOT NULL default '0',
  `comment` text NOT NULL default "",
  `format` tinyint(4) unsigned NOT NULL default '0',
  `teacher` int(10) unsigned NOT NULL default '0',
  `timemarked` int(10) unsigned NOT NULL default '0',
  `mailed` tinyint(1) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `assignment` (`assignment`),
  KEY `userid` (`userid`),
  KEY `mailed` (`mailed`),
  KEY `timemarked` (`timemarked`)
) COMMENT='Info about submitted assignments';
# -----

INSERT INTO prefix_log_display (module, action, mtable, field) VALUES ('assignment', 'view', 'assignment', 'name');
INSERT INTO prefix_log_display (module, action, mtable, field) VALUES ('assignment', 'add', 'assignment', 'name');
INSERT INTO prefix_log_display (module, action, mtable, field) VALUES ('assignment', 'update', 'assignment', 'name');
INSERT INTO prefix_log_display (module, action, mtable, field) VALUES ('assignment', 'view submission', 'assignment', 'name');
INSERT INTO prefix_log_display (module, action, mtable, field) VALUES ('assignment', 'upload', 'assignment', 'name');

```

Access.php

```
<?php
$mod_assignment_capabilities = array(
    'mod/assignment:view' => array(
        'capytype' => 'read',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'guest' => CAP_ALLOW,
            'student' => CAP_ALLOW,
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'admin' => CAP_ALLOW
        )
    ),
    'mod/assignment:submit' => array(
        'capytype' => 'write',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'student' => CAP_ALLOW
        )
    ),
    'mod/assignment:grade' => array(
        'capytype' => 'write',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'admin' => CAP_ALLOW
        )
    )
);
?>
```

Anexo 2. Código fuente módulo “Ejemplo”

Fichero mod.html

```

<?php // $Id: mod.html,v 1.5 2006/10/07 12:28:57 gustav_delius Exp $
/**
 * This page defines the form to create or edit an instance of this module
 * It is used from /course/mod.php. The whole instance is available as $form.
 *
 * @author
 * @version $Id: mod.html,v 1.5 2006/10/07 12:28:57 gustav_delius Exp $
 * @package NEWMODULE
 **/

/// First we check that form variables have been initialised

if (!isset($form->name)) {
    $form->name = "";
}
if (!isset($form->description)) {
    $form->description = "";
}
if (!isset($form->grade)) {
    $form->grade = 100;
}

// More similar blocks go here...
?>

<form name="form" method="post" action="mod.php">
<center>
<table cellpadding="5">
<tr valign="top">
    <td align="right"><b><?php print_string("name", "ejemplo") ?></b></td>
    <td>
        <input type="text" name="name" size=60 value="<?php p($form->name) ?>">
    </td>
</tr>
<tr valign="top">
    <td align="right"><b><?php print_string("description", "ejemplo") ?></b>
    </td>
    <td><?php
print_textarea($usehtmleditor, 20, 60, 680, 400, "description", $form->description);

if ($usehtmleditor) {
    echo '<input type="hidden" name="format" value=".FORMAT_HTML." />';
} else {
    echo '<div align="right">';
    helpbutton("textformat", get_string("formattexttype"));
    print_string("formattexttype");
    echo ':&nbsp;';
    if (!$form->format) {
        $form->format = $defaultformat;
    }
    choose_from_menu(format_text_menu(), "format", $form->format, "");
    echo '</div>';
}
?>
</tr>
<tr valign="top">
    <td align="right"><b><?php print_string("grade", "ejemplo") ?></b></td>
    <td>
        <?php
print_grade_menu($course->id, "grade", "$form->grade");

```

```

?>
</td>
</tr>
<!-- More rows go in here... -->

<!-- The following line for Moodle 1.5 prints the visibility setting form element -->
<?php print_visible_setting($form); ?>
<!-- and if your module uses groups you would also have -->
<?php print_groupmode_setting($form); ?>

</table>
<!-- These hidden variables are always the same -->
<input type="hidden" name=course      value="<?php p($form->course) ?>" />
<input type="hidden" name=module      value="<?php p($form->module) ?>" />
<input type="hidden" name=modulename  value="<?php p($form->modulename) ?>" />
<input type="hidden" name=instance   value="<?php p($form->instance) ?>" />
<input type="hidden" name=sesskey     value="<?php p($form->sesskey) ?>" />
<input type="hidden" name="mode"     value="<?php p($form->mode) ?>" />
<input type="hidden" name="section"   value="<?php p($form->section) ?>" />
<input type="submit" value="<?php print_string("savechanges") ?>" />
</center>

</form>

```

Fichero Mysql.sql

```

# $Id: mysql.sql,v 1.2 2006/08/28 16:41:20 mark-nielsen Exp $
# This file contains a complete database schema for all the
# tables used by this module, written in SQL

# It may also contain INSERT statements for particular data
# that may be used, especially new entries in the table log_display

CREATE TABLE `prefix_ejemplo` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `course` int(10) unsigned NOT NULL default '0',
  `name` varchar(255) NOT NULL default '',
  `description` text NOT NULL,
  `grade` int(10) NOT NULL default '0',
  `timemodified` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`)
) COMMENT='Define Ejemplo';

```

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or

processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- * A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- * B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- * C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

-
- * D. Preserve all the copyright notices of the Document.
 - * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
 - * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
 - * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - * H. Include an unaltered copy of this License.
 - * I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - * K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - * M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - * N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.