

## Ampliación de Moodle: Creación de módulo actividad

Raúl Ivorra Oltra, Sergio Luján Mora.

*Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante.  
Apartado de Correos 99, E-03080 Alicante, España  
rio@alu.ua.es, slujan@dlsi.ua.es.*

### Resumen

La plataforma educativa Moodle se está extendiendo en los últimos años casi de forma exponencial y está despertando mucho interés. Uno de los puntos fuertes de esta plataforma es la capacidad de expansión, junto con su carácter de plataforma abierta.

A la hora de plantearse Moodle como una posible plataforma de aprendizaje, uno de los factores que se deben analizar son los módulos disponibles y si sus características se amoldan a las necesidades a cubrir en el proceso de enseñanza/aprendizaje. En Moodle también existe la posibilidad de crear nuestros propios módulos, y es esta posibilidad la que se va a explorar en este artículo.

La documentación disponible para el desarrollo de nuevos módulos es escasa, así como la información general sobre la arquitectura del sistema, sin embargo la documentación del código es muy completa. El objetivo de este artículo es proporcionar las bases para el desarrollo de un módulo, en concreto se explicarán las pautas para la creación de una actividad.

**Palabras Clave:** Moodle; actividad; LMS.

### Abstract

The Moodle learning platform is expanding in recent years almost exponentially and is attracting much interest. One of the strengths of this platform is the ability to expand, along with its open platform.

When Moodle is evaluated as a possible learning platform, one of the factors that are discussed are the modules available and if their characteristics are tailored to the needs to cover in the process of teaching / learning. In Moodle, you have the option of creating your own modules, and this is the possibility we explore in this paper.

The documentation available for the development of new modules is limited, and general information about the system architecture is also limited, however the documentation of the code is very complete. The aim of this paper is to provide the basis for the development of a module, specifically this paper explains the steps for the creation of an activity.

**Keywords:** Moodle; activity; LMS.

## 1. Introducción

Moodle es un sistema de gestión de aprendizaje o LMS (*Learning Management System*) creado por Martin Dougiamas [1, 2] bajo licencia de software libre. La primera versión

de Moodle data de 2002 y desde entonces goza de un fuerte crecimiento que es prácticamente exponencial [3]. Moodle está traducido a más de 75 idiomas [1].

En diferentes entornos docentes el uso de Moodle ha despertado un gran interés y ello se ha visto reflejado en materiales docentes y ponencias a congresos educativos [4-7].

Entre las principales características de Moodle se encuentra su enfoque pedagógico basado en el constructivismo, mediante el uso de la colaboración, las actividades, y la promoción de la reflexión crítica [2]. Los requisitos para su instalación son mínimos, ya que solamente requiere de un servidor web con soporte para PHP (idealmente la versión 5 para aprovechar la orientación a objetos de este lenguaje de programación [8]) y un sistema gestor de base de datos, soportando los más comunes en el entorno del software libre como son MySQL y PostgreSQL, aunque soporta otros.

Moodle nos proporciona una interfaz sencilla, ligera, eficiente y compatible con los estándares y es apropiado tanto para cursos exclusivamente online, como para complementar los cursos presenciales mediante una formación combinada (*blended learning*).

Moodle dispone de unos módulos ya integrados muy flexibles y que cubren la gran mayoría de las necesidades de una plataforma de aprendizaje actual, como son: tareas, chat, consulta, foro, cuestionario, recursos, encuesta y taller [8].

Moodle como plataforma modular (de ahí la M de Moodle) es ampliable mediante extensiones (*pulgins*), que pueden ser módulos de actividad, bloques, filtros, métodos de autenticación, entre otros. Al ser software libre también es posible colaborar con el desarrollo y mejora de la plataforma base, y acceder al código de ésta.

Pese a que es posible añadir funcionalidades, la plataforma base es bastante completa de por sí. Además, el creador y desarrollador principal ha creado una empresa dedicada a prestar servicios sobre la plataforma como desarrollos a medida, instalaciones, alojamiento, etc.; los frutos de estos servicios se van añadiendo a la plataforma, quedando disponibles para la comunidad de usuarios.

El código de la plataforma está bien documentado [9, 10], lo que permite localizar y entender el funcionamiento de la plataforma fácilmente. No obstante, no existe apenas

información general de la arquitectura de la plataforma, ni de cómo interactúan las distintas clases y librerías que lo componen. La única forma de hacerse una idea de la arquitectura es mediante el estudio del código de algunas de las partes de la plataforma.

El objetivo de este artículo es proporcionar las bases para el desarrollo de un módulo, en concreto se explicarán las pautas y pasos a seguir para la creación de una actividad. Para ello se explicará la estructura general de un módulo actividad, así como las particularidades de Moodle a tener en cuenta en el desarrollo.

El resto del artículo se estructura en los siguientes apartados: en el apartado 2, Esquema de un módulo, se describe la estructura de directorios y archivos que se emplea para almacenar un módulo de Moodle; en el apartado 3, Desarrollo de un módulo, se explica paso a paso cómo desarrollar un módulo nuevo para Moodle; en el apartado 4, Facilidades de la plataforma, se citan algunas de las librerías y de los mecanismos existentes en Moodle que pueden ayudar a desarrollar un nuevo módulo; en el apartado 5, Inconvenientes y dificultades, se comentan los principales problemas enfrentados durante el desarrollo del módulo explicado en este artículo; y, por último, en el apartado 6, Conclusiones, se presentan las conclusiones y los trabajos futuros.

## 2. Esquema de un módulo.

Previo al desarrollo de una actividad es necesario conocer el esquema de un módulo de Moodle. Los módulos se almacenan en la carpeta `moodle/mod` de la instalación de Moodle, cada uno en un directorio individual, siendo la estructura general de directorios y archivos como resumimos a continuación (en la Fig. 1 se resume la estructura de un módulo):

- `mod_form.php`: Formulario para crear o modificar una instancia de la actividad. Fig. 1 (q).
- `version.php`: Contiene meta información sobre el módulo, como por ejemplo la versión del módulo. Fig. 1 (r).

- `lang/`: Directorio para almacenar los archivos de idioma del módulo y de ayuda. El módulo debe tener archivos de idioma que contenga las cadenas para ese módulo. Deberán ser al menos en inglés y traducidos a los idiomas de los usuarios finales que utilicen la actividad. Fig. 1 (e).

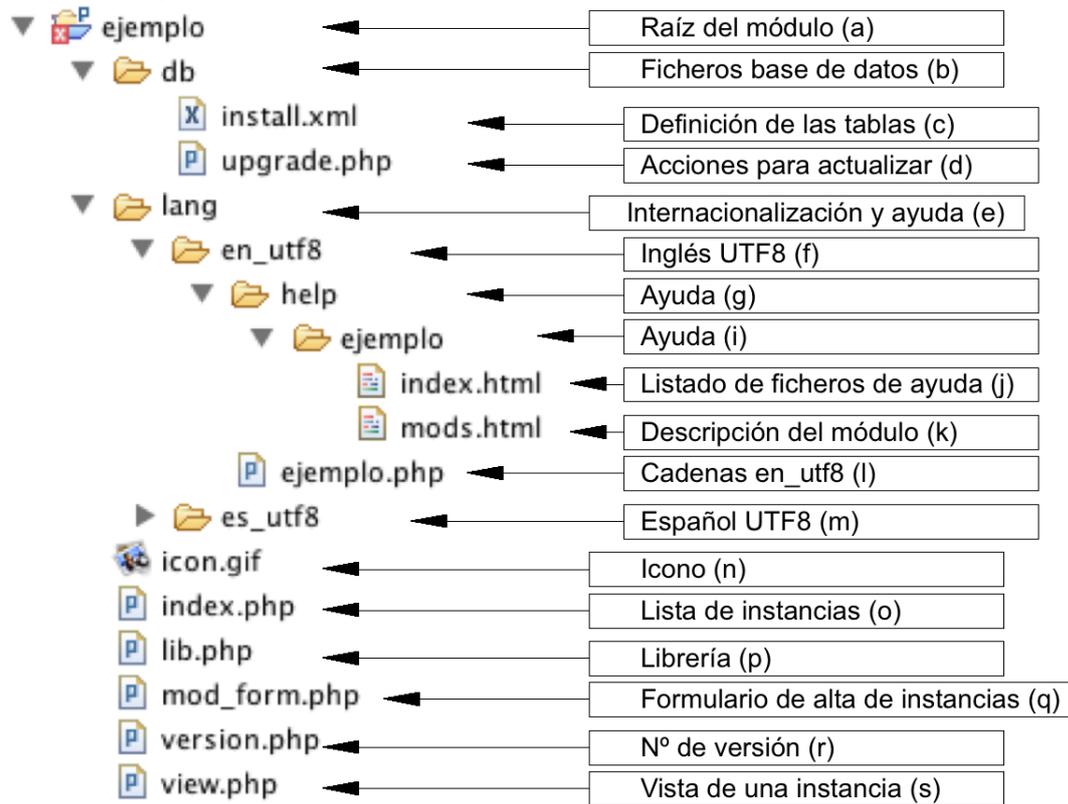
La estructura de carpetas y archivos para cada idioma es la siguiente:

- `en_utf8/`: Directorio para los ficheros de idioma y ayuda en inglés. Fig. 1 (f).
  - `newmodule.php`: Fichero de idioma del módulo *newmodule*. Fig. 1 (l).
  - `help/`: Carpeta para los ficheros de ayuda en el idioma inglés. Fig. 1 (g).
    - `newmodule/`: Carpeta donde se ubican los ficheros de ayuda. Fig. 1 (i). Los obligatorios son:
      - `index.html`: Listado de los ficheros de ayuda. Fig. 1 (j).
      - `mods.html`: Descripción general del módulo. Fig. 1 (k).
- `db/`: Directorio donde se almacenarán los ficheros con las tablas de las bases de datos necesarias para la actividad. Fig. 1 (b).
- `access.php`: Fichero opcional que contiene los permisos del módulo. Los permisos no son obligatorios, pero sí muy recomendables para garantizar que usuarios pueden acceder a las distintas partes del módulo.
- `install.xml`: Fichero que describe la estructura de las tablas del módulo. Fig. 1 (c).
- `upgrade.php`: Código de actualización, aquí es donde se deben de hacer las alteraciones de las tablas, si las hay, entre versiones. Fig. 1 (d).
- `index.php`: Este fichero sirve para mostrar todas las instancias de una actividad en un curso, es decir, una lista con todas las instancias del mismo módulo. Fig. 1 (o).

- `view.php`: Ésta es la página que muestra una instancia de la actividad. Fig. 1 (s).
- `lib.php`: Librería de funciones del módulo. Fig. 1 (p). En este fichero se implementarán todas las funciones y procedimientos del módulo. Si el módulo se llama *ejemplo*, entonces las funciones mínimas y obligatorias que ha de tener la actividad tienen que ser de la forma:
  - *ejemplo\_install()*: Acciones a realizar al instalar el módulo.
  - *ejemplo\_add\_instance()*: Código para añadir una nueva instancia.
  - *ejemplo\_update\_instance()*: Función para actualizar una instancia existente.
  - *ejemplo\_delete\_instance()*: Código para borrar una instancia.
  - *ejemplo\_user\_outline()*: Da un resumen concreto de la actividad de un usuario.
  - *ejemplo\_user\_complete()*: Devuelve un informe más detallado de la contribución de un usuario.
  - *ejemplo\_get\_view\_actions()*: Clasifica las acciones para el log. Se usa en el informe de participación.

Todas las funciones, procedimientos y constantes, creados en `lib.php`, tienen que comenzar con el nombre del módulo.

- `settings.php`: Fichero opcional que contiene el formulario con las opciones generales del módulo.



**Figura 1:** Esquema de carpetas de un módulo actividad.

### 3. Desarrollo de un módulo.

A continuación describimos el proceso de desarrollo de un módulo de actividad para Moodle. Explicaremos el proceso de creación de nuevas instancias, la creación y uso de las tablas de un módulo, y también se repasarán los ficheros clave de un módulo. Debido a las limitaciones de espacio, la explicación de este apartado es un resumen, una explicación más detallada se puede encontrar en nuestro tutorial [11].

En primer lugar hay que descargarse la plantilla para un nuevo módulo de [12]. Esta plantilla tiene todos los ficheros obligatorios, y la estructura de directorios ya comentada en el apartado anterior.

Los pasos básicos para realizar un módulo son:

- Desarrollo del formulario para configurar cada nueva instancia del módulo.

- Creación de las tablas editando el fichero `db/install.xml`. Fig. 1 (c).
- Edición de la librería de funciones del archivo `lib.php`. Fig. 1 (p).
- Edición del resto de archivos obligatorios para adecuarlos al nuevo módulo: `index.php`, `version.php` y ficheros de idiomas. Fig. 1 (o) y (r).
- Edición del fichero `view.php` que muestra cada instancia. Fig. 1 (s).

A continuación se describen uno por uno todos estos pasos.

### 3.1 Desarrollo del formulario para configurar cada nueva instancia del módulo

El fichero `mod_form.php` (Fig. 1 (q)) es llamado cada vez que se añade una instancia de una actividad y con él la configuraremos. En la plantilla ya tenemos los campos obligatorios como *name* y otros típicos como *intro*. Cambiaremos *newmodule* por el nombre de nuestro módulo, con el prefijo *mod\_* delante del nombre del módulo en el nombre de la clase.

Aquí nos serviremos de la modularidad que nos ofrece la plataforma para no preocuparnos de cómo guardar los datos en la base de datos ni de cómo se define un formulario en HTML, simplemente con extender la clase *moodlemod\_form* como esta en la plantilla y añadir los campos de configuración de la instancia que definamos en la base de datos es suficiente.

A continuación podremos añadir elementos al formulario de los que encontramos en la documentación [13]. Para finalizar se añadirán los botones estándar para permitir procesar el formulario.

### 3.2 Creación de las tablas del módulo.

Todo módulo necesita de al menos una tabla para guardar las propiedades de cada instancia, la estructura de las tablas se define en el fichero `install.xml`. Fig. 1 (c)

La plantilla ya proporciona un fichero válido en el que sólo hay que modificar el nombre de la tabla y algunos comentarios. Una vez instalado el módulo ya podremos editar las tablas desde Moodle.

La sintaxis de este fichero no es muy sencilla como para editar el fichero directamente a mano; por eso se recomienda editarlo desde la plataforma con un usuario administrador en el menú *Miscelánea*, opción *Editor XMLDB*, evitando los errores del editado manual.

### 3.3 Edición de los ficheros obligatorios

El resto de ficheros obligatorios que debemos de editar son:

- `lib.php`: Todo módulo ha de tener una librería con las funciones para el manejo de las instancias. En este fichero se definen también las constantes y clases que pueda necesitar el módulo. Sustituiremos *newmodule* por el nombre de nuestro módulo en todos los sitios oportunos siguiendo las instrucciones que nos proporciona el propio fichero de la plantilla para preparar las funciones necesarias.
- `index.php`: Solamente con cambiar *newmodule* por el nombre de nuestro módulo ya listará todas las instancias del mismo.
- `version.php`: Indicamos la fecha que identifica a la versión. El valor que utilizemos lo tendremos en cuenta en el fichero `db/upgrade.php`. Fig. 1 (d).
- Ficheros de idioma: Siempre debemos incluir el inglés para facilitar la traducción a otros idiomas. Los ficheros han de estar en la ruta `lang/en_utf8/ejemplo.php` (Fig. 1 (l)) y en castellano en la ruta `lang/es_utf8/ejemplo.php`.

### 3.4 Edición del fichero `view.php`

Este fichero es quizás el más importante de un módulo ya que muestra la instancia, es decir, muestra la nueva actividad.

En la plantilla, como en los anteriores, sustituyendo *newmodule* por el nombre de nuestro módulo tendremos la base para crear nuestra actividad ya que en una variable tendremos todos los valores de la instancia. A partir de ahí, es tarea del desarrollador proporcionarle la funcionalidad deseada a la actividad.

#### **4. Facilidades de la plataforma.**

En la creación de un módulo podemos y debemos usar las librerías disponibles en Moodle. Éstas se encuentran en el directorio `lib` de la instalación. Decimos debemos principalmente porque nos ahorrarán trabajo y porque ya se encuentran integradas en la plataforma, por lo que su uso es muy fácil.

A continuación citamos algunas de las librerías que son imprescindibles en el uso en un módulo y algunas que hemos considerado interesantes:

- **ADODB:** Esta librería es básica, ya que nos proporciona acceso a la base de datos de forma independiente del sistema gestor de base de datos que estemos utilizando. Con esta librería podemos realizar las operaciones básicas de consulta, actualización y borrado, aunque también podemos ejecutar sentencias SQL en caso de necesidad.
- **XMLDB:** Esta librería se utiliza para definir las tablas que se crean en la instalación de un módulo. Al igual que la librería anterior, nos permite crear las tablas independientemente del gestor de base de datos de la instalación.
- **PEAR (*PHP Extension and Application Repository*):** Algunas de las librerías PHP de este repositorio están disponibles en la instalación. Nos permiten trabajar con los elementos HTML como por ejemplo los formularios mediante la librería Quickform. Con ésta librería creamos, validamos y procesamos formularios HTML.

- YUI (*Yahoo User Interface Library*): Esta librería es un ejemplo del tipo de librerías que podemos encontrar en la instalación por defecto. En concreto esta librería libre desarrollada por Yahoo nos permite crear interfaces basados en JavaScript.

Como recomendación final de este apartado, antes de afrontar el desarrollo de un módulo recomendamos revisar las librerías disponibles ya que todas ellas son libres y nos permiten centrarnos en la funcionalidad a añadir en vez de en cómo añadirla.

## 5. Inconvenientes y dificultades encontradas.

El mayor inconveniente ha sido el de no disponer de documentación acerca de la arquitectura de Moodle. No existe documentación actualmente acerca de cómo se organizan los datos, ni de cómo interactúan entre sí las clases y librerías que incorpora. Esa información ha de deducirse de la documentación del código [9, 10], que sí que es muy completa.

Como resultado de la falta de documentación citada anteriormente, hemos encontrado dificultades para crear un módulo con una funcionalidad básica que no utilice una combinación de HTML, PHP y/o Javascript. Es decir, si al desarrollar un módulo nuevo para Moodle nos limitamos a utilizar solamente las clases y librerías integradas en el sistema, sin tener que recurrir al uso de código HTML o JavaScript directamente, la dificultad es mucho mayor.

## 6. Conclusiones

La plataforma Moodle permite su ampliación en muchos aspectos [14] mediante la creación de módulos para esta. No obstante, como es una plataforma de aprendizaje que experimenta un fuerte crecimiento desde su lanzamiento en 2002 [3], es conveniente revisar sus funcionalidades y librerías ya incorporadas antes de embarcarse en la creación de una nueva.

Debido a la falta de documentación de la arquitectura de Moodle, el desarrollo de un módulo nuevo para Moodle puede ser complicado, por lo que se recomienda consultar la documentación del código [9, 10].

Como complemento a este artículo, se puede consultar en [11] un tutorial acerca de la creación de un módulo actividad, con información mucho más detallada y que pretendemos que sirva de base para la iniciación en el desarrollo de módulos actividad para Moodle.

## 7. Referencias

1. Wikipedia. *Moodle*. Internet: <http://es.wikipedia.org/wiki/Moodle>. [Visitado Abril 2009]
2. Moodle. *Acerca de Moodle*. Internet: [http://docs.moodle.org/es/Acerca\\_de\\_Moodle](http://docs.moodle.org/es/Acerca_de_Moodle). [Visitado Abril 2009]
3. Moodle. *Moodle Statistics*. Internet: <http://moodle.org/stats/>. [Visitado Marzo 2009]
4. J.V. Pruñonosa Reverter y R. Reverter Forcadell, *MoodleMoot 2008*, Desarrollo de un curso entre dos profesores mediante Moodle, 2008.
5. A. Gámez Mellado, L.M. Marín Trechera, S. Fandiño Patiño, R. Rodríguez Huertas, *XVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (CUIEET)*, Integración LAMS-Moodle: Diversas Experiencias en Asignaturas en el Campus Virtual de la Universidad de Cádiz, 2008.
6. C. Latorre Castillo, J.C. Fernández Llibre, S. Bernadés Tel, *XVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (CUIEET)*, Implementación de la plataforma Moodle en la EUSS, 2008.
7. H. Trochanová, *9th International Conference Virtual University (VU'08)*, The e-learning course Informatics in LMS Moodle, 2008.
8. Moodle. *Características de Moodle*. Internet: <http://docs.moodle.org/es/Características>. [Visitado Abril 2009]

9. Moodle. *Moodle Technical Documentation (phpdocumentor)*. Internet: <http://phpdocs.moodle.org/>. [Visitado Marzo 2009]
10. Moodle. *PHP Cross Reference of Moodle 1.9.3*. Internet: <http://xref.moodle.org/>. [Visitado Marzo 2009]
11. R. Ivorra Oltra. *Tutorial: Creación de un módulo actividad Moodle (1.9.3)*. Internet: <http://gpsl.dlsi.ua.es/~slujan/materiales/creacion-modulo-moodle.pdf>. [Visitado Febrero 2009]
12. Moodle. *Plantilla para nuevo módulo*. Internet: <http://download.moodle.org/plugins/mod/NEWMODULE.zip>. [Visitado Marzo 2009]
13. Moodle. *Development: lib/formslib.php Form Definition*. Internet: [http://docs.moodle.org/en/Development:lib/formslib.php\\_Form\\_Definition](http://docs.moodle.org/en/Development:lib/formslib.php_Form_Definition). [Visitado Marzo 2009]
14. Moodle. *How you can contribute to Moodle*. Internet: [http://docs.moodle.org/en/Development:Developer\\_documentation#How\\_you\\_can\\_contribute](http://docs.moodle.org/en/Development:Developer_documentation#How_you_can_contribute). [Visitado Abril 2009]