

# GUÍA DE APOYO PARA EL USO DE



## 1.9.4

# Usuario Desarrollador



Documento creado por Ana Teresa González de Felipe.

La siguiente guía de usuario está basada en documentos similares, los cuales se mencionan en el apartado final “Referencias Bibliográficas”, así como en la experiencia propia obtenida de la instalación, uso y desarrollo de la plataforma Moodle.

Esta guía se distribuye bajo Licencia de Documentación Libre de GNU, sin restricciones adicionales. Es libre de copiar, distribuir, y modificarse este texto según los términos indicados por dicha licencia. El texto completo de la licencia puede consultarse en la siguiente dirección web: <http://www.gnu.org/copyleft/fdl.html>.



## Guías de usuario

---

En este apartado del documento se pretende proporcionar una visión global de Moodle que le permita entender su estructura y comportamiento general para, más adelante, poder ahondar en los conocimientos sobre la plataforma dependiendo del uso que el usuario quiera darle a la misma.

Visto desde fuera, Moodle es un sitio web, con soporte para el registro de usuarios, en los que cada usuario puede adoptar un rol que le permite interactuar de distintas maneras con la propia herramienta Moodle o con el resto de usuarios.

Una primera idea sobre Moodle es concebirlo como algo similar al sistema de enseñanza tradicional, en el que un año lectivo consta de varias asignaturas (los cursos) estructuradas en semanas o temas que constan de varias actividades de aprendizaje. Además, como en toda educación escolar, existen dos papeles básicos, el de profesor, creador del contenido del curso, propulsor de las actividades, etc., y el de alumno, la persona que recibirá el conocimiento, realizará las actividades propuestas y, finalmente, será evaluado.



# Índice

<b>1</b>	<b>ESTRUCTURA BÁSICA DE MOODLE</b>	<b>13</b>
1.1	CATEGORÍAS	14
1.2	CURSOS	14
1.3	SEMANAS Y TEMAS	14
1.4	ACTIVIDADES EN MOODLE	15
1.5	RECURSOS EN MOODLE	21
1.6	BLOQUES EN MOODLE	22
1.7	ROLES, CAPACIDADES Y CONTEXTOS EN MOODLE	22
1.8	LOS GRUPOS EN MOODLE	24
1.9	EL SISTEMA DE EVALUACIÓN EN MOODLE	24
<b>2</b>	<b>GUÍA DE APOYO PARA EL USUARIO DESARROLLADOR</b>	<b>25</b>
2.1	INTRODUCCIÓN	27
2.2	INSTALACIÓN DEL ENTORNO DE TRABAJO	29
2.2.1	<i>Obtención de un entorno de trabajo</i>	29
2.2.2	<i>Instalación de Apache2Triad en una máquina con SO Windows XP</i>	29
2.2.3	<i>Instalación de Moodle 1.9.2 en una máquina con SO Windows XP</i>	33
2.2.4	<i>Instalación de Xampp en una máquina con SO Windows Vista</i>	35
2.2.5	<i>Instalación de Moodle 1.9.2 en una máquina con SO Windows Vista</i>	39
2.2.6	<i>Instalación de Moodle en una máquina con SO Mac OS X 10.5 Servidor</i>	40
2.2.7	<i>Instalación de Xampp en una máquina con SO Ubuntu</i>	45
2.2.8	<i>Instalación de Moodle en una máquina con SO Ubuntu</i>	47
2.3	MANUAL DE ESTILO DE CÓDIGO	49
2.3.1	<i>Reglas generales</i>	49
2.3.2	<i>Estilo del código</i>	50
2.3.3	<i>Estructura de la base de datos</i>	52
2.3.4	<i>Normas de seguridad (y control de la información de formularios y URLs)</i>	54
2.4	MANUAL DE ESTILO DE INTERFAZ	56
2.5	CVS PARA DESARROLLADORES	59
2.5.1	<i>Unirse al proyecto como desarrollador</i>	59
2.5.2	<i>Módulos CVS</i>	59
2.5.3	<i>Comandos básicos</i>	60
2.5.4	<i>Trabajando con ramas</i>	62
2.6	DESARROLLO DE UN MÓDULO DE ACTIVIDADES EN MOODLE	65
2.6.1	<i>Estructura de un módulo en Moodle</i>	65
2.6.2	<i>Librerías de Moodle para el desarrollo de módulos</i>	66
2.7	FICHEROS DE UN MÓDULO DE ACTIVIDAD	75
2.7.1	<i>El fichero INSTALL.XML</i>	75
2.7.2	<i>El fichero ACCESS.PHP</i>	76
2.7.3	<i>El fichero MOD.HTML</i>	77
2.7.4	<i>El fichero VIEW.PHP</i>	78
2.7.5	<i>El fichero LIB.PHP</i>	79
2.7.6	<i>El fichero INDEX.PHP</i>	81
2.7.7	<i>El fichero VERSION.PHP</i>	82
2.7.8	<i>El fichero UPGRADE.PHP</i>	83
2.7.9	<i>Los ficheros de idioma</i>	83

2.7.10	<i>Los ficheros de Ayuda</i> .....	84
<b>3</b>	<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>87</b>

# Índice de ilustraciones

Figura 1 Esquema de la composición de Moodle .....	13
Figura 320 Usuario desarrollador: Panel de control de Apache2Triad .....	30
Figura 321 Usuario desarrollador: phpMyAdmin.....	31
Figura 322 Usuario desarrollador: crear nueva base de datos .....	32
Figura 323 Usuario desarrollador: instalación de Moodle 1 .....	33
Figura 324 Usuario desarrollador: instalación de Moodle 2 .....	33
Figura 325 Usuario desarrollador: instalación de Moodle 3 .....	34
Figura 326 Usuario desarrollador: instalación de Moodle 4 .....	34
Figura 327 Usuario desarrollador: instalación de Moodle 5 .....	35
Figura 328 Usuario desarrollador: instalación de Moodle 6 .....	35
Figura 329 Usuario desarrollador: panel de control de XAMPP .....	36
Figura 330 Usuario desarrollador: página principal de XAMPP en el servidor local .....	37
Figura 331 Usuario desarrollador: phpMyAdmin.....	38
Figura 332 Usuario desarrollador: crear nueva base de datos .....	39
Figura 333 Usuario desarrollador: phpInfo1 en Mac .....	42
Figura 334 Usuario desarrollador: phpinfo2 en Mac.....	42
Figura 335 Usuario desarrollador: phpMyAdmin en Mac .....	43
Figura 336 Usuario desarrollador: página principal del curso en mac.....	44
Figura 337 Usuario desarrollador: página principal de Xampp para Linux en el servidor local.....	46
Figura 338 Usuario desarrollador: phpInfo en Linux.....	46
Figura 339 Usuario desarrollador: phpMyAdmin en Linux.....	47
Figura 340 Usuario desarrollador: phpInfo en Linux.....	47
Figura 341 Usuario desarrollador: ramas de Moodle.....	62



# Índice de tablas

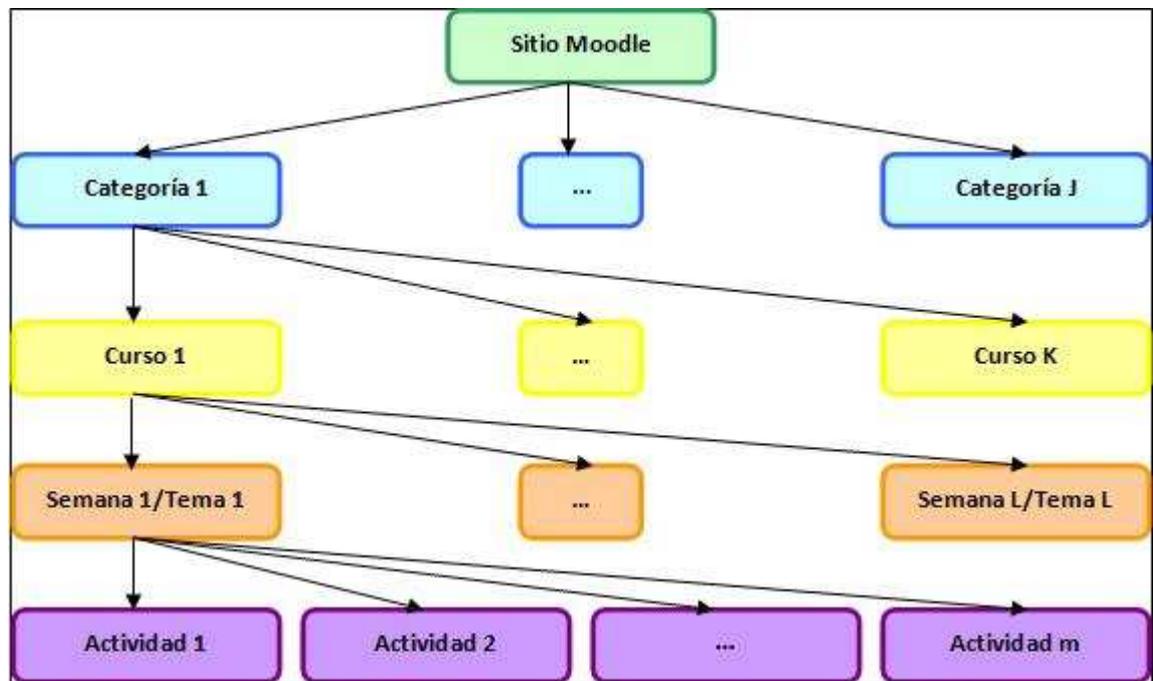
---

<b>Tabla 1</b> Lista de actividades por defecto de Moodle.....	<b>15</b>
<b>Tabla 2</b> Lista de recursos en Moodle .....	<b>21</b>
<b>Tabla 3</b> Lista de algunos bloques en Moodle .....	<b>22</b>
<b>Tabla 4</b> Lista de contextos en Moodle .....	<b>23</b>
<b>Tabla 5</b> Lista de roles en Moodle .....	<b>24</b>
<b>Tabla 10</b> Usuario desarrollador: listado y descripción de librerías y funciones básicas.....	<b>67</b>



# 1 Estructura básica de Moodle

Un sitio Moodle está compuesto por: categorías, cursos, temas (o semanas) y actividades.



*Figura 1 Esquema de la composición de Moodle*

A continuación se procederá a explicar cada uno de los elementos que completan un sitio Moodle.

## 1.1 Categorías

Las categorías son los contenedores de información de más alto nivel, están formadas por cursos y sirven para organizarlos de manera que sean más fácilmente localizables por el alumno en la pantalla inicial de la aplicación.

## 1.2 Cursos

Los cursos son la parte más importante de la estructura de Moodle, son creados por los administradores del sitio y dirigidos por los usuarios que se establezcan como profesores del curso.

Para que un alumno, dado de alta en el sistema, pueda acceder a un curso deberá estar matriculado en él.

Cuando un administrador crea un curso debe proceder a su configuración mediante un formulario proporcionado por Moodle en el que se establecen valores para distintos campos como, por ejemplo:

- Nombre y descripción.
- Formato del curso (semanal, por temas...)
- Número de semanas o temas.
- Fechas en las que permanecerá abierto el curso.

Una vez que un curso es creado, la matriculación la puede llevar a cabo cada usuario de forma independiente o bien el propio administrador, de forma masiva, sobre un grupo de usuarios del sistema, definiendo qué usuarios serán profesores y cuales alumnos.

## 1.3 Semanas y Temas

La organización de un curso podrá llevarse a cabo por semanas o bien por temas, según la preferencia del profesorado.

Tras la creación del curso, su configuración, y la matriculación de usuarios, estos podrán acceder al mismo y observar una serie de bloques diferenciados que representan las semanas del curso o temas, según el formato que se haya establecido. Cada uno de estos bloques contendrá, a partir del momento en el que el profesor las añada, distintos tipos de actividades que los alumnos deberán realizar para su evaluación.

## 1.4 Actividades en Moodle

Moodle ofrece la posibilidad de añadir cada semana o tema distintos tipos de actividades. A continuación se muestran aquellas disponibles con la plataforma Moodle.

*Tabla 1 Lista de actividades por defecto de Moodle*

LISTA DE ACTIVIDADES POR DEFECTO DE MOODLE		
Nombre	Descripción	Características
Base de Datos 	Permite que los usuarios incorporen datos desde un formulario diseñado por el profesor.	<ul style="list-style-type: none"> <li>• Permite crear una base de datos accesible, en lectura y escritura, tanto al alumnado como al profesorado.</li> <li>• Tiene diferentes tipos de campos: <ul style="list-style-type: none"> <li>• Texto</li> <li>• Imágenes</li> <li>• Archivo</li> <li>• URL</li> <li>• Fecha</li> <li>• Menú</li> <li>• Menú (Selección múltiple)</li> <li>• Botón de marcar (Checkbox)</li> <li>• Botones de elección (Radio buttons)</li> </ul> </li> </ul>
Chat 	Permite conversaciones entre usuarios en tiempo real.	<ul style="list-style-type: none"> <li>• Permite una interacción fluida mediante texto síncrono.</li> <li>• Incluye las fotos de los perfiles en la ventana de chat.</li> <li>• Soporta direcciones URL, emoticones, integración de HTML, imágenes, etc.</li> <li>• Todas las sesiones quedan registradas para verlas posteriormente, y pueden ponerse a disposición de los estudiantes.</li> </ul>
Consulta 	Similar a una encuesta, el	<ul style="list-style-type: none"> <li>• Es como una votación. Puede usarse</li> </ul>

	<p>profesor formula una única pregunta y ofrece a los usuarios distintas elecciones.</p>	<p>para votar sobre algo o para recibir una respuesta de cada estudiante (por ejemplo, para pedir su consentimiento para algo).</p> <ul style="list-style-type: none"> <li>• El profesor puede ver una tabla que presenta de forma intuitiva la información sobre quién ha elegido qué.</li> <li>• Se puede permitir que los estudiantes vean un gráfico actualizado de los resultados.</li> </ul>
<p>Cuestionario </p>	<p>Permite la realización de exámenes de diferente tipo: respuesta múltiple, verdadero/falso y respuestas cortas.</p>	<ul style="list-style-type: none"> <li>• Los profesores pueden definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios.</li> <li>• Las preguntas pueden ser almacenadas en categorías de fácil acceso, y estas categorías pueden ser "publicadas" para hacerlas accesibles desde cualquier curso del sitio.</li> <li>• Los cuestionarios se califican automáticamente, y pueden ser recalificados si se modifican las preguntas.</li> <li>• Los cuestionarios pueden tener un límite de tiempo a partir del cual no estarán disponibles.</li> <li>• El profesor puede determinar si los cuestionarios pueden ser resueltos varias veces y si se mostrarán o no las respuestas correctas y los comentarios.</li> <li>• Las preguntas y las respuestas de los cuestionarios pueden ser mezcladas (aleatoriamente) para disminuir las copias entre los alumnos.</li> <li>• Las preguntas pueden crearse en HTML y con imágenes.</li> <li>• Las preguntas pueden importarse</li> </ul>

		<p>desde archivos de texto externos.</p> <ul style="list-style-type: none"> <li>• Los intentos pueden ser acumulativos, y acabados tras varias sesiones.</li> <li>• Las preguntas de opción múltiple pueden definirse con una única o múltiples respuestas correctas.</li> <li>• Pueden crearse preguntas de respuesta corta (palabras o frases).</li> <li>• Pueden crearse preguntas tipo verdadero/falso.</li> <li>• Pueden crearse preguntas de emparejamiento.</li> <li>• Pueden crearse preguntas aleatorias.</li> <li>• Pueden crearse preguntas numéricas (con rangos permitidos).</li> <li>• Pueden crearse preguntas de respuesta incrustada (estilo "cloze") con respuestas dentro de pasajes de texto.</li> <li>• Pueden crearse textos descriptivos y gráficos.</li> </ul>
<p>Encuesta </p>	<p>Similar a la consulta, pero con varias preguntas.</p>	<ul style="list-style-type: none"> <li>• Se proporcionan encuestas ya preparadas (COLLES, ATTLS) y contrastadas como instrumentos para el análisis de las clases en línea.</li> <li>• Los informes de las encuestas están siempre disponibles, incluyendo muchos gráficos. Los datos pueden descargarse con formato de hoja de cálculo Excel o como archivo de texto CVS.</li> <li>• La interfaz de las encuestas impide la posibilidad de que sean respondidas sólo parcialmente.</li> <li>• A cada estudiante se le informa sobre sus resultados comparados con la media de la clase.</li> </ul>

<p>Foro </p>	<p>Actividad para el debate entre usuarios de un curso.</p>	<ul style="list-style-type: none"> <li>• Hay diferentes tipos de foros disponibles: exclusivos para los profesores, de noticias del curso y abiertos a todos.</li> <li>• Todos los mensajes llevan adjunta la foto del autor.</li> <li>• Las discusiones pueden verse anidadas, por rama, o presentar los mensajes más antiguos o el más nuevo primero.</li> <li>• El profesor puede obligar la suscripción de todos a un foro o permitir que cada persona elija a qué foros suscribirse de manera que se le envíe una copia de los mensajes por correo electrónico.</li> <li>• El profesor puede elegir que no se permitan respuestas en un foro (por ejemplo, para crear un foro dedicado a anuncios).</li> <li>• El profesor puede mover fácilmente los temas de discusión entre distintos foros.</li> <li>• Las imágenes adjuntas se muestran dentro de los mensajes.</li> <li>• Si se usan las calificaciones de los foros, pueden restringirse a un rango de fechas.</li> </ul>
<p>Glosario </p>	<p>Permite crear y mantener una lista de definiciones, como un diccionario.</p>	<ul style="list-style-type: none"> <li>• Muestra en su interior la definición o descripción de cualquiera de los glosarios que se haya definido en el curso.</li> </ul>
<p>Lección </p>	<p>Consiste en una serie de páginas flexibles que ofrecen contenido y preguntas para el alumno.</p>	<ul style="list-style-type: none"> <li>• Al final de cada página se plantea una pregunta con varias posibles respuestas.</li> <li>• Según la opción que escoja el alumno para esa respuesta se le mostrarán unas u otras de las páginas restantes.</li> </ul>
<p>Recurso</p>	<p>Admite la presentación de cualquier contenido digital,</p>	<ul style="list-style-type: none"> <li>• Los archivos pueden subirse y manejarse en el servidor, o pueden ser</li> </ul>

	Word, PowerPoint, Flash, vídeo, sonidos, etc.	<p>creados sobre la marcha usando formularios Web (de texto o HTML).</p> <ul style="list-style-type: none"> <li>• Se pueden enlazar contenidos externos en Web o incluirlos perfectamente en la interfaz del curso.</li> <li>• Pueden enlazarse aplicaciones Web, transfiriéndoles datos.</li> </ul>
SCORM 	Bloque de material Web empaquetado siguiendo el estándar SCORM de objetos de aprendizaje	<ul style="list-style-type: none"> <li>• Este bloque puede incluir páginas Web, gráficas, programas JavaScript, presentaciones Flash y cualquier otra cosa que funcione en un navegador Web.</li> <li>• Permite cargar fácilmente cualquier paquete SCORM (Sharable Content Object Reference Model) estándar y convertirlo en parte de un curso.</li> </ul>
Taller	Actividad de trabajo en grupo que permite la evaluación entre estudiantes.	<ul style="list-style-type: none"> <li>• Permite la evaluación de documentos entre iguales, y el profesor puede gestionar y calificar la evaluación.</li> <li>• Admite un amplio rango de escalas de calificación posibles.</li> <li>• El profesor puede suministrar documentos de ejemplo a los estudiantes para practicar la evaluación.</li> <li>• Es muy flexible y tiene muchas opciones.</li> </ul>
Tarea 	Permiten al profesor calificar trabajos enviados por los alumnos.	<ul style="list-style-type: none"> <li>• Puede especificarse la fecha final de entrega de una tarea y la calificación máxima que se le podrá asignar.</li> <li>• Los estudiantes pueden subir sus tareas (en cualquier formato de archivo) al servidor. Se registra la fecha en que se han subido.</li> <li>• Se permite enviar tareas fuera de tiempo, pero el profesor puede ver claramente el tiempo de retraso.</li> <li>• Para cada tarea en particular, puede</li> </ul>

		<p>evaluarse a la clase entera (calificaciones y comentarios) en una única página con un único formulario.</p> <ul style="list-style-type: none"><li>• Las observaciones del profesor se adjuntan a la página de la tarea de cada estudiante y se le envía un mensaje de notificación.</li><li>• El profesor tiene la posibilidad de permitir el reenvío de una tarea tras su calificación (para volver a calificarla).</li></ul>
Wiki 	Posibilita la creación colectiva de documentos en un lenguaje simple de marcas utilizando un navegador Web.	<ul style="list-style-type: none"><li>• Permite a los participantes trabajar juntos en páginas Web para añadir, expandir o modificar su contenido.</li><li>• Las versiones antiguas nunca se eliminan y pueden restaurarse.</li></ul>

## 1.5 Recursos en Moodle

Para complementar el contenido de las semanas o temas de cada curso, Moodle, además de actividades, ofrece la posibilidad de agregar otro tipo de recursos a las mismas:

*Tabla 2 Lista de recursos en Moodle*

LISTA DE RECURSOS EN MOODLE	
Nombre	Descripción
Editar una página de texto 	Añade un enlace a un texto plano creado por el profesor.
Editar una página Web 	Añade un enlace a un documento creado con formato HTML por el profesor.
Enlazar un archivo o una página Web 	Permite añadir un enlace desde el curso a cualquier sitio Web público.
Directorio 	Muestra directorios, subdirectorios y archivos del área de archivos del curso.
Desplegar paquetes de contenido IMS 	<p>IMS es un esqueleto de especificaciones que ayuda a definir variados estándares técnicos, incluyendo materiales de e-learning. La especificación IMS (Content Packaging Specification) hace posible almacenar los contenidos en un formato estándar que puede ser reutilizado en diferentes sistemas sin necesidad de convertir dichos contenidos a otros formatos.</p> <p>El recurso de tipo IMS CP en Moodle, le permite utilizar ese tipo de paquetes de contenidos, cargándolos e incluyéndolos en cursos de forma sencilla.</p>
Añadir una etiqueta 	Permite colocar texto e imágenes entre otras actividades de la página central de un curso

## 1.6 Bloques en Moodle

Las funcionalidades extra fuera de las actividades y recursos son implementadas por los bloques, contenedores que se sitúan a los lados del sitio Web y que tienen una función concreta de carácter general (relacionada o no con los módulos de actividades):

*Tabla 3 Lista de algunos bloques en Moodle*

ALGUNOS BLOQUES EN MOODLE	
Nombre	Descripción
Calendario	Muestra un calendario con los eventos próximos en el curso.
Canales RSS externos	Permite la lectura de canales RSS desde el sitio Moodle.
HTML	Muestra determinado contenido HTML introducido por el usuario.
Descripción del curso	Muestra la descripción introducida durante la creación o configuración del curso.

## 1.7 Roles, capacidades y contextos en Moodle

Para diferenciar entre tipos de usuarios, Moodle proporciona un sistema de roles en función de los permisos y capacidades que deban asignarse a cada usuario, entendiendo por capacidad a la posibilidad de realizar cierta acción en el sistema. Así, un estudiante tendrá, entre otras, capacidades para realizar actividades, mientras que un profesor tendrá capacidades para la edición de un curso y un administrador las necesarias para introducir cualquier tipo de modificaciones en el sitio Web.

Los contextos sirven para establecer distintos niveles de capacidades, están ordenados de forma jerárquica, así, si un usuario tiene cierta capacidad en el nivel más alto de la jerarquía, heredarán estos permisos en los niveles que estén por debajo, a continuación se listan los contextos existentes en Moodle:

*Tabla 4 Lista de contextos en Moodle*

CONTEXTOS EN MOODLE	
Nombre	Descripción
CONTEXT_SYSTEM	Capacidades definidas a nivel del sitio completo
CONTEXT_PERSONAL	Capacidades referentes al usuario actual
CONTEXT_USER	Capacidades referentes a los usuarios
CONTEXT_COURSECAT	Capacidades referentes a todos los cursos de una categoría
CONTEXT_COURSE	Capacidades referentes a los curso
CONTEXT_GROUP	Capacidades referentes a un grupo de trabajo
CONTEXT_MODULE	Capacidades referentes a un módulo de actividades
CONTEXT_BLOCK	Capacidades referentes a un bloque

Ya se han nombrado los roles de profesor, estudiante y administrador, aunque en Moodle existen muchos más, de hecho infinitos, ya que, en las más recientes versiones de la plataforma, el propio administrador puede definir roles y asignarles distintas capacidades. No obstante, existen ciertos roles por defecto preestablecidos en Moodle:

*Tabla 5 Lista de roles en Moodle*

ROLES EN MOODLE	
Nombre	Descripción
Primary Admin	Administrador primario o principal
Admins	Administradores
Course Creators	Creadores del curso
Editing Teachers	Profesores con capacidad de editar
Non-Editing Teachers	Profesores sin capacidad de editar
Students	Estudiantes
Guest	Invitados

A los roles listados anteriormente, un administrador puede asignarles las capacidades que determine oportunas.

## 1.8 Los grupos en Moodle

Actualmente, los grupos en Moodle presentan cierta analogía con los grupos establecidos en un curso real (presencial), en el que cada grupo está formado por un conjunto de estudiantes que acuden a una sesión de teoría común, esta funcionalidad no tiene utilidad más allá de poder separar a los alumnos para que accedan a unas u otras tareas, no sirve para fomentar el trabajo en grupo, en contra de lo que su nombre pueda sugerir en un principio.

## 1.9 El sistema de evaluación en Moodle

Todo el sistema de actividades y roles sobre el que se sustenta la filosofía de aprendizaje de Moodle no tendría sentido si no existiera una manera de evaluar y calificar a los usuarios que desempeñan el papel de estudiantes. Así, la mayoría de actividades presentan la opción de establecer una nota numérica a cada alumno, de manera que éste pueda consultarlas en cualquier momento.

Moodle incluye un sistema de evaluación común a todas las actividades evaluables del sitio (incluso las actividades desarrolladas por otros usuarios externos a Moodle), de manera que tanto un profesor como un alumno puede conocer el estado de las calificaciones en cada actividad del curso de manera individual y la nota media de todas ellas, ponderando, si se desea, el peso correspondiente a cada actividad.

## 2 Guía de apoyo para el usuario desarrollador

Este documento no pretende ser unas normas cerradas, pero sí deben tenerse en cuenta las sugerencias que la propia comunidad Moodle hace para desarrollar módulos.



## 2.1 Introducción

A continuación se presenta una guía para todo aquel que quiera iniciarse en el desarrollo Moodle, en ella se detallarán los diferentes tipos de contribuciones que un desarrollador puede hacer a la comunidad Moodle.

Desde la perspectiva de un administrador de sistemas, Moodle ha sido diseñado de acuerdo con los siguientes criterios:

### 1. Moodle debe poder ejecutarse en la más amplia posible variedad de plataformas

La plataforma de aplicaciones Web que funciona en la mayoría de las plataformas es PHP combinada con MySQL, y este es el entorno en el que Moodle ha sido desarrollado (sobre Linux, Windows, y Mac OS X). Moodle también usa la librería ADOdb para la abstracción de bases de datos, lo que significa que Moodle puede usar más de diez marcas diferentes de bases de datos (desafortunadamente, a pesar de ello, no puede aún crear tablas en todas esas bases de datos. Hablaremos más sobre esto más adelante).

### 2. Moodle debe ser fácil de instalar, aprender y modificar

Los primeros prototipos de Moodle (1999) se construyeron usando Zope, un avanzado servidor de aplicaciones Web orientado a objetos. Desafortunadamente me pareció que aunque la tecnología era bastante buena, tenía una curva de aprendizaje muy elevada y no era muy flexible en términos de administración del sistema. El lenguaje PHP, por otro lado, es muy fácil de aprender (especialmente si has hecho algo de programación usando cualquier otro lenguaje de script). Pronto tomé la decisión de evitar usar un diseño orientado a clases, con la finalidad, una vez más, de mantenerlo fácil de entender para los principiantes. La reutilización del código se archiva en librerías con funciones claramente tituladas y con una disposición de los archivos de script, consistente. PHP es también fácil de instalar (existen versiones ejecutables para todas las plataformas) y está ampliamente disponible, pues la mayoría de los servicios de alojamiento lo proporcionan como un estándar

### 3. Moodle debe ser fácil de actualizar desde una versión hasta la siguiente

Moodle sabe cuál es su versión (así como las versiones de todos los módulos) y se ha construido un mecanismo interno para que Moodle pueda actualizarse a sí mismo de forma apropiada a las nuevas versiones (por ejemplo, puede renombrar las tablas de las bases de datos o añadir nuevos campos). Usando CVS en Unix, por ejemplo, uno tan sólo tiene que hacer un "cvs update -d" y luego visitar la página principal del sitio para completar la actualización.

### 4. Moodle debe ser modular para permitir el crecimiento

Moodle tiene una serie de características modulares, incluyendo temas, actividades, interfaces de idioma, esquemas de base de datos y formatos de cursos. Esto le permite a cualquiera añadir características al código básico principal o incluso distribuirlas por separado. Hablaremos más de esto en la siguiente sección.

### 5. Moodle debe poder usarse junto a otros sistemas

Una de las cosas que hace Moodle es mantener todos los archivos para un curso en un único directorio en el servidor. Esto podría permitir que el administrador de un sistema proporcione similares formas de acceso a un nivel de archivo para cada profesor, tal como Appletalk, SMB, NFS, FTP, WebDAV y demás. Los módulos de autenticación le permiten a Moodle usar LDAP, IMAP, POP3, NNTP y otras bases de datos como fuentes de información de los usuarios. Por otra parte, aún queda trabajo por hacer sobre esto. Para futuras versiones de Moodle tenemos planeadas las siguientes características: importación y exportación de los datos de Moodle utilizando formatos basados en XML (incluyendo IMS y SCORM), incrementar el uso de hojas de estilo para el formateo de interfaces (de manera que puedan integrarse visualmente en otros sitios Web).

## 2.2 Instalación del entorno de trabajo

Los pasos que aquí se detallan servirán para llegar a configurar un entorno válido y así, posteriormente, obtener una instalación satisfactoria de Moodle en el equipo. Estos pasos están pensados para todos aquellos usuarios que utilicen la plataforma para desarrollar módulos.

Es conveniente decir que la explicación que viene a continuación está indicada para un sistema operativo Windows XP y Windows Vista y otra más compacta para los sistemas operativos Mac Os y Linux.

### 2.2.1 Obtención de un entorno de trabajo

Al ser **Moodle un proyecto de software libre desarrollado en PHP**, será necesario que en nuestra máquina tengamos instalado un **intérprete PHP** y un **servidor web**. Además, Moodle trabaja con **bases de datos** (recomendando el propio equipo de desarrollo de Moodle que se utilice MySQL).

Por todo esto, se aconseja utilizar cualquiera de los paquetes existentes en la red que contienen estos elementos básicos y además se encuentran de forma gratuita. Algunos de estos paquetes son: Xampp, Apache2Triad, EasyPHP, AppServ, PHPTriad...

A continuación se tomará como ejemplo la instalación Apache2Triad en una máquina con un sistema operativo Windows XP y Xampp para otra máquina con Windows Vista.

### 2.2.2 Instalación de Apache2Triad en una máquina con SO Windows XP

1. El paquete se puede descargar de la siguiente página: <http://apache2triad.net/> y una vez aquí ir a la opción Downloads.

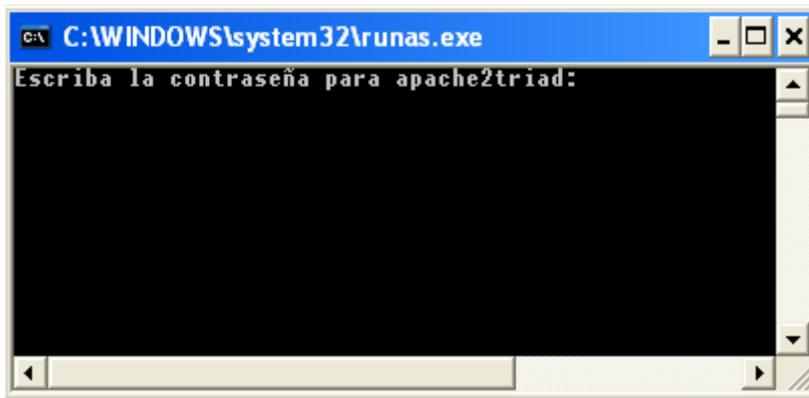


En este caso la versión escogida ha sido la última hasta el momento, Apache2Triad 1.5.4

2. Una vez descargado el ejecutable en la máquina en la que se va a trabajar, es necesario descomprimirla. En este proceso se solicitará una contraseña que posteriormente se pedirá. (Se recomienda que a partir de este



momento se utilice la misma contraseña, para así evitar problemas).



3. Una vez acabado el proceso le pedirá reiniciar el ordenador para poder hacer efectivos los cambios realizados

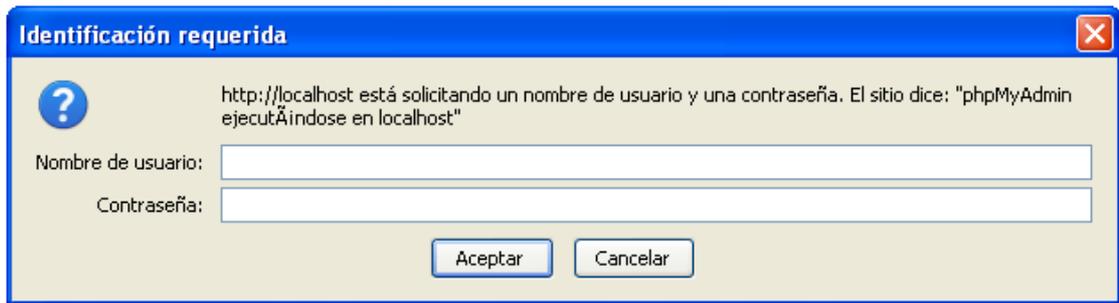


4. Reiniciado el ordenador debe entrar en el navegador y escribir <http://localhost/phpMyAdmin>. Esto le llevará al servidor local recién instalado. (Podría ser que le diera error y esto se deberá a que no están activados los módulos antes descritos (apache y MySQL). Para que funcione correctamente comprobar que Apache2Triad Manager tiene el siguiente aspecto accediendo a través de inicio/todos los programas/Apache2Triad/ Apache2TriadManager).



**Figura 2** Usuario desarrollador: Panel de control de Apache2Triad

- Si todo ha ido bien le pedirá un nombre de usuario, deberá escribir “root” y como contraseña la misma que escribió durante la instalación



- La página a la que accederá por medio de la dirección antes mencionada tiene el siguiente aspecto y deberá entrar en la opción “Privilegios”.

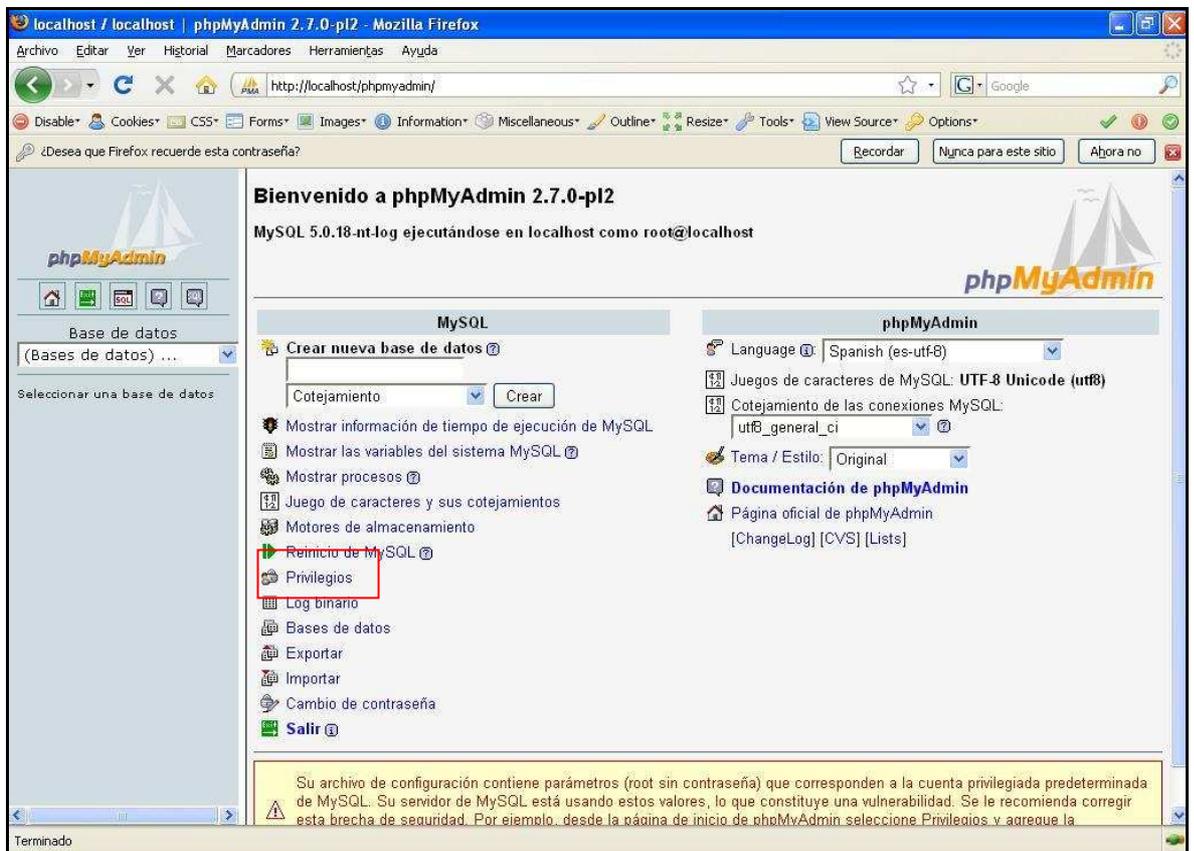


Figura 3 Usuario desarrollador: phpMyAdmin

- En la tabla que se despliega, en la fila que cuyo usuario es “root”, haga clic en editar como se muestra a continuación:

	Usuario	Servidor	Contraseña	Privilegios globales	Conceder	
<input type="checkbox"/>	cualquiera	%	--	USAGE	No	
<input type="checkbox"/>	root	localhost	No	ALL PRIVILEGES	Sí	

[Marcar todos/as / Desmarcar todos](#)

8. Establecer una contraseña para el “root”.



Cambio de contraseña

Sin contraseña

Contraseña:

Debe volver a escribir:

Continuar

9. En este momento es necesario ir a c:\apache2triad\htdocs\phpmyadmin y editar el archivo “config.inc.php”.

10. Modificar en ...['user']='root'; cambiar ['password'] ='' ; y poner la contraseña (la misma que se puso en el punto 8), posteriormente guardar (existen dos lugares en el archivo donde hay que modificarlo y la forma en que aparece es la siguiente).

```
$cfg['Servers'][$i]['user'] = 'root';  
$cfg['Servers'][$i]['password'] = '';
```

11. En este momento ya sólo queda crear la base de datos. Para ello, debe volver a la página principal pinchando en el icono:



12. Haga clic sobre “Crear nueva base de datos” y escriba “moodle”, como se muestra a continuación:



MySQL

Crear nueva base de datos

moodle Cotejamiento

Crear

Figura 4 Usuario desarrollador: crear nueva base de datos

13. Si todo ha ido correctamente le mostrará la información siguiente:

```
Base de datos moodle se creó.  
consulta SQL:  
CREATE DATABASE `moodle`;
```

14. Ya está listo para instalar la plataforma MOODLE.

## 2.2.3 Instalación de Moodle 1.9.2 en una máquina con SO Windows XP

Hasta el momento de crear esta documentación, la última versión existente de la plataforma es la 1.9.2 y será sobre la que se muestre la explicación.

1. Descargue Moodle de la siguiente dirección: <http://download.moodle.org/>
2. Una vez que tenga el archivo comprimido en su ordenador, descomprímalo en la dirección donde tenga Apache2Triad. Si se ha utilizado la dirección por defecto estará en c:\apache2triad\htdocs (es necesario puntualizar que la carpeta htdocs es en la que se buscarán todos los archivos a ejecutar cuando se utilice el servidor local)
3. A partir de este punto comienza la instalación en su máquina, para ello es necesario entrar en nuestro navegador y escribir la dirección <http://localhost/moodle>
4. Seleccione el idioma en el que desea que continúe la instalación, en este ejemplo se elige el español (hay varias opciones de español, en este caso se escoge el español de España), pulse sobre "next". En este momento la instalación sigue en el idioma que haya seleccionado.



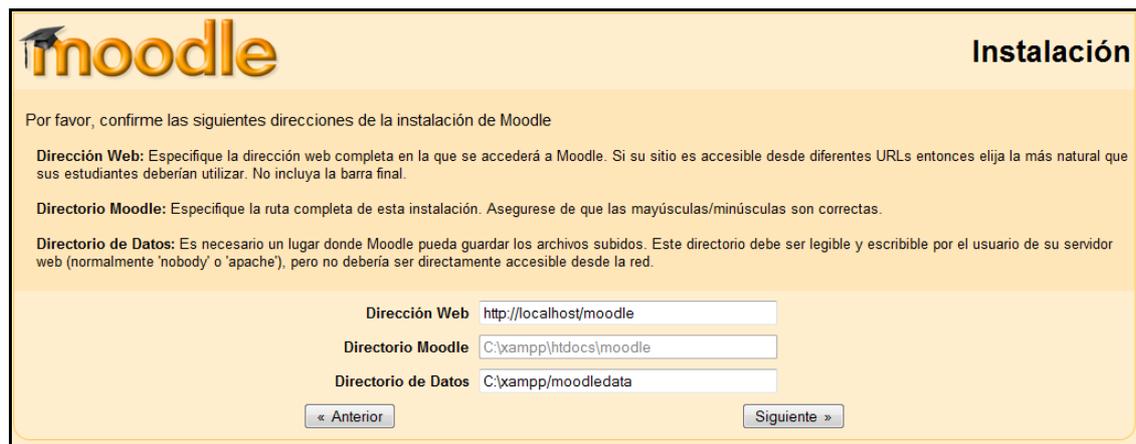
**Figura 5 Usuario desarrollador: instalación de Moodle 1**

5. Compruebe que todo está correcto en los ajustes del servidor.



**Figura 6 Usuario desarrollador: instalación de Moodle 2**

6. Datos a tener en cuenta en la pantalla siguiente:



Por favor, confirme las siguientes direcciones de la instalación de Moodle

**Dirección Web:** Especifique la dirección web completa en la que se accederá a Moodle. Si su sitio es accesible desde diferentes URLs entonces elija la más natural que sus estudiantes deberían utilizar. No incluya la barra final.

**Directorio Moodle:** Especifique la ruta completa de esta instalación. Asegurese de que las mayúsculas/minúsculas son correctas.

**Directorio de Datos:** Es necesario un lugar donde Moodle pueda guardar los archivos subidos. Este directorio debe ser legible y escribible por el usuario de su servidor web (normalmente 'nobody' o 'apache'), pero no debería ser directamente accesible desde la red.

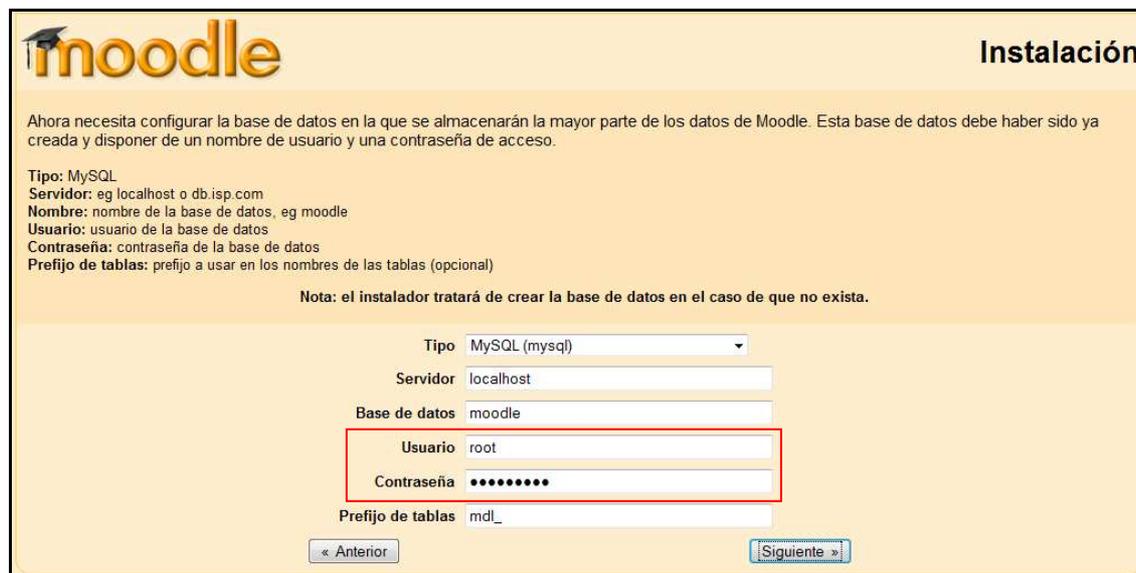
Dirección Web

Directorio Moodle

Directorio de Datos

**Figura 7 Usuario desarrollador: instalación de Moodle 3**

7. En la siguiente ventana introducir como usuario: "root" y como contraseña la que pusiera en PHPMyAdmin y pulse "Siguiente".



Ahora necesita configurar la base de datos en la que se almacenarán la mayor parte de los datos de Moodle. Esta base de datos debe haber sido ya creada y disponer de un nombre de usuario y una contraseña de acceso.

Tipo: MySQL  
Servidor: eg localhost o db.isp.com  
Nombre: nombre de la base de datos, eg moodle  
Usuario: usuario de la base de datos  
Contraseña: contraseña de la base de datos  
Prefijo de tablas: prefijo a usar en los nombres de las tablas (opcional)

Nota: el instalador tratará de crear la base de datos en el caso de que no exista.

Tipo

Servidor

Base de datos

Usuario

Contraseña

Prefijo de tablas

**Figura 8 Usuario desarrollador: instalación de Moodle 4**

8. Las comprobaciones del servidor es recomendable que estén en **OK**. En caso de que aparezca alguna que ponga "revisar" procure ir a donde le indica y modifíquelo. Pulse "Siguiente".

9. Ya casi al final de la instalación le solicitará descargar el paquete de idioma. **Para hacer esta opción es necesario estar conectado a Internet.** Otra opción para el idioma es descargarlo posteriormente desde la dirección [http://download.moodle.org/lang16/es\\_es.zip](http://download.moodle.org/lang16/es_es.zip) y descomprimirlo en "C:\Apache2triad\htdocs\moodle\lang". Pulse "Siguiente".

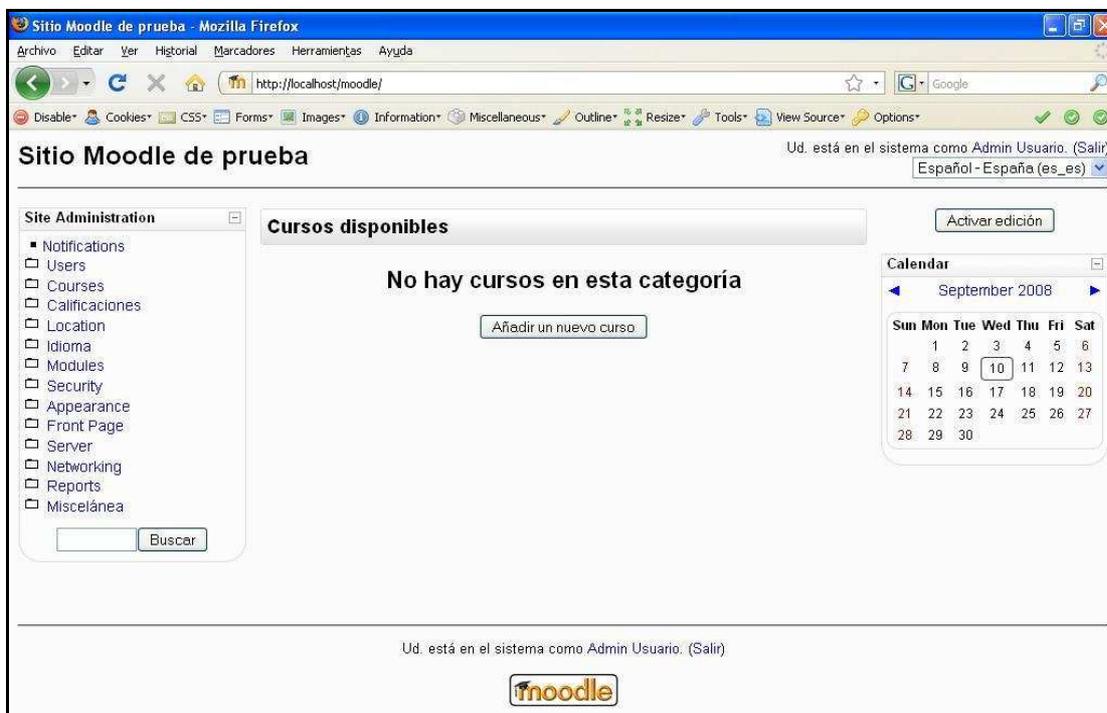
El idioma "Español - España (es\_es)" ha sido instalado correctamente. El proceso de instalación continuará en este idioma.

10. Acepte las condiciones de uso y pulse "Siguiente".
11. En este momento se instalarán las bases de datos. Sólo debe comprobar que todo ha ido correctamente y pinchar en "continuar"



**Figura 9 Usuario desarrollador: instalación de Moodle 5**

12. Ahora le pedirán los datos del administrador del sitio.
13. Introduzca también el nombre del sitio Moodle que va a utilizar. En este punto ya tiene instalada la plataforma Moodle en su máquina siendo la página principal del sitio la siguiente:



**Figura 10 Usuario desarrollador: instalación de Moodle 6**

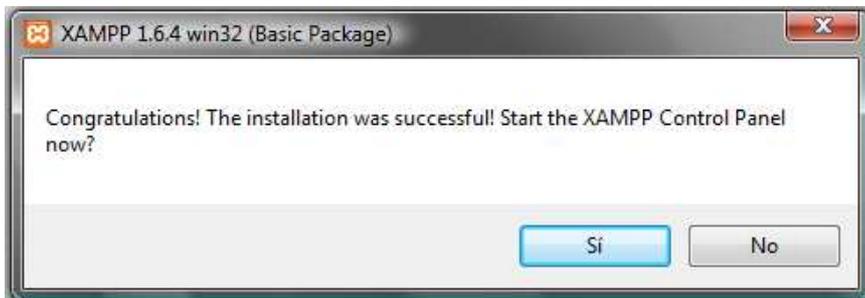
## 2.2.4 Instalación de Xampp en una máquina con SO Windows Vista

1. Descargar el paquete de la siguiente dirección: <http://www.apachefriends.org/en/xampp.html>, ir a Xampp for Windows→Download Xampp→ Installer.
2. Una vez tenga en su máquina el instalador, ejecútelo. Lo primero que le pedirá es seleccionar el

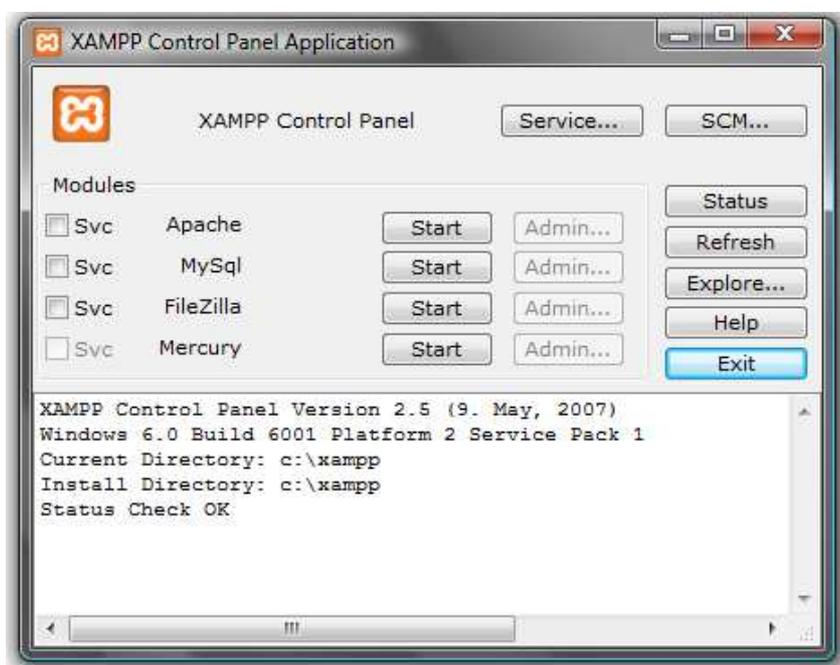


idioma para la ejecución: las opciones son inglés, alemán y chino. En este ejemplo se solicita inglés.

3. Seguir los pasos marcados por el propio instalador pulsando “Next” hasta llegar a “Finish”.



4. Llegado a este punto se despliega el siguiente panel de control de Xampp, donde deberá pulsar “Start” en “Apache” y en MySQL (En principio esto sólo deberá hacerlo la primera vez que acceda al panel de control de XAMPP).

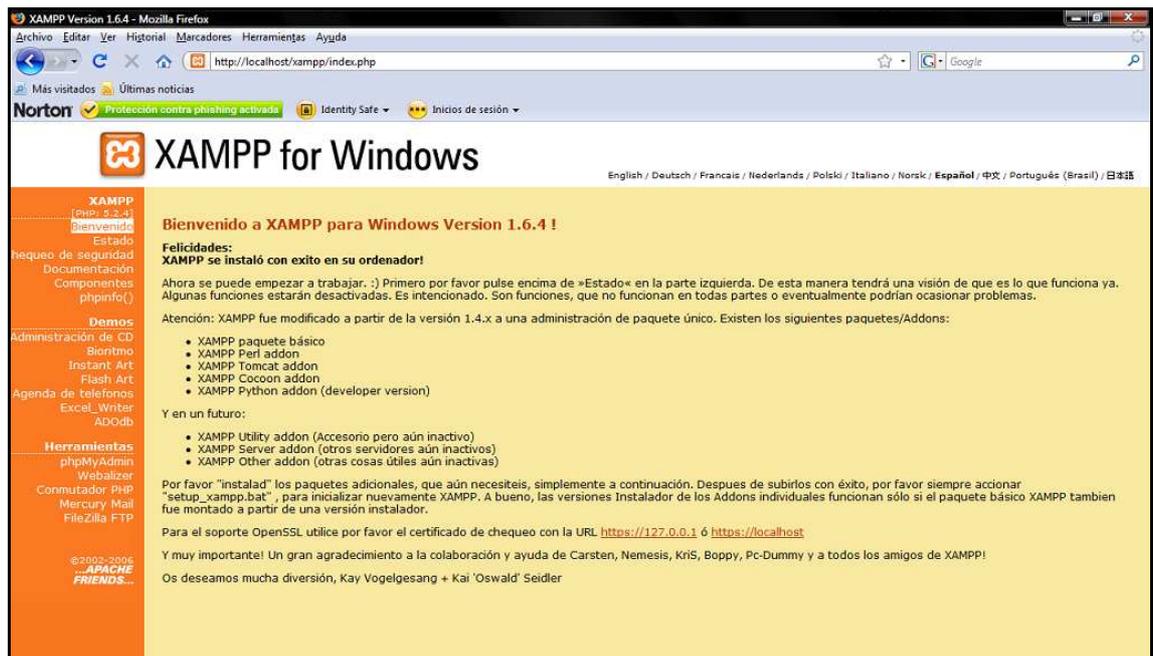


**Figura 11 Usuario desarrollador: panel de control de XAMPP**

5. Vaya al navegador y escriba la siguiente dirección: <http://localhost>, se le abrirá una página con la siguiente apariencia:

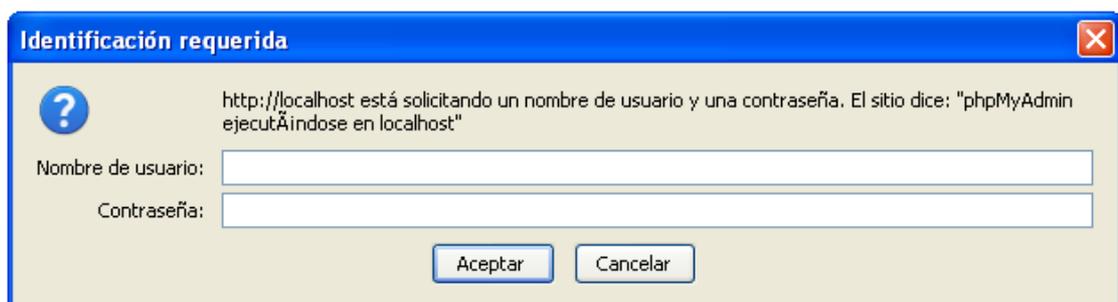


6. Seleccione el idioma, en este caso español y le llevará a la página principal del sitio local.



**Figura 12** Usuario desarrollador: página principal de XAMPP en el servidor local

7. Haga clic en PHPMyAdmin (situado en el menú de Herramientas de la izquierda) o escriba en la barra de direcciones: <http://localhost/phpmyadmin> .
8. Si todo ha ido bien le pedirá un nombre de usuario, deberá escribir “root” y como contraseña dejar el campo vacío.



9. La página a la que accederá por medio de la dirección antes mencionada tiene el siguiente aspecto y deberá entrar en la opción “Privilegios”.

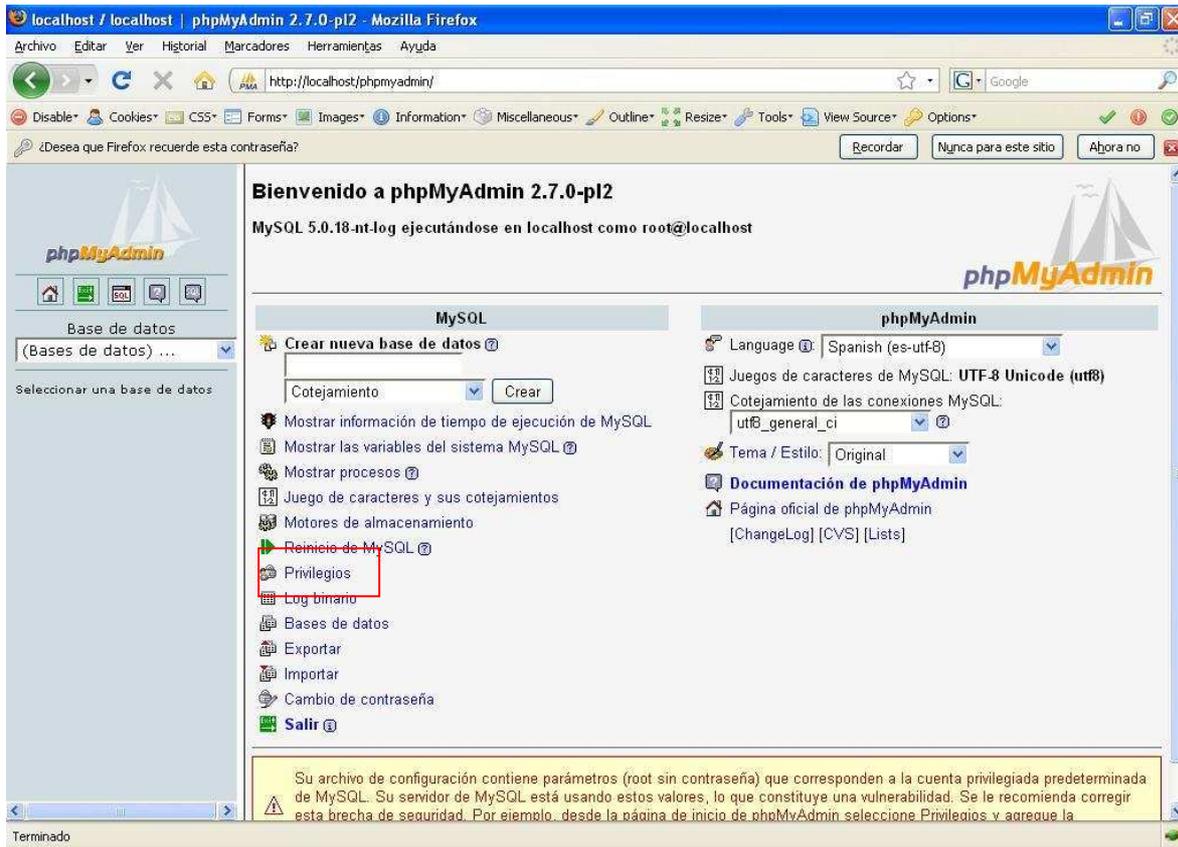


Figura 13 Usuario desarrollador: phpMyAdmin

10. En la tabla que se despliega, en la fila que cuyo usuario es “root”, haga clic en editar como se muestra a continuación:

	Usuario	Servidor	Contraseña	Privilegios globales	Conceder	
<input type="checkbox"/>	cualquiera	%	--	USAGE	No	
<input type="checkbox"/>	root	localhost	No	ALL PRIVILEGES	Sí	

[Marcar todos/as / Desmarcar todos](#)

11. Establecer una contraseña para el “root”.

Cambio de contraseña

Sin contraseña  
 Contraseña:   
 Debe volver a escribir:

12. Vaya al archivo c:/xampp/phpMyAdmin/config.inc.php, edítelo.

```
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = '';
```

Modifique lo siguiente:

```
$cfg['Servers'][$i]['user'] = 'root'; // MySQL user
$cfg['Servers'][$i]['password'] = 'nuevacontraseña'; // MySQL password (only needed
// with 'config' auth_type)
```

13. En este momento ya sólo queda crear la base de datos. Para ello, debe volver a la página principal pinchando en el icono:

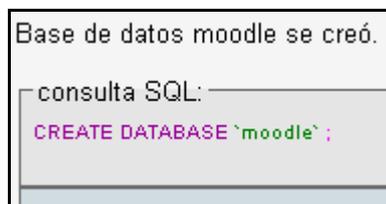


14. Haga clic sobre “Crear nueva base de datos” y escriba “moodle”, como se muestra a continuación:



Figura 14 Usuario desarrollador: crear nueva base de datos

15. Si todo ha ido correctamente le mostrará la información siguiente:



16. Llegado a este punto ya tiene instalado el paquete Xampp en su equipo.

## 2.2.5 Instalación de Moodle 1.9.2 en una máquina con SO Windows Vista

Hasta el momento de crear esta documentación, la última versión existente de la plataforma es la 1.9.2 y será sobre la que se trabaje en esta explicación.

1. Descargue Moodle de la siguiente dirección: <http://download.moodle.org/>
2. Descomprima Moodle en la dirección: c:\xampp\htdocs
3. Entre en <http://127.0.0.1/moodle> (es importante entrar en el sitio local como **127.0.0.1**, ya que si se intenta hacer a través de localhost le dará errores en la instalación y no llegará a completarse la instalación).

4. A partir de este aquí se pueden seguir los pasos como en el [punto 4](#) y siguientes de la instalación de Moodle en un Sistema Operativo Windows XP.

## 2.2.6 Instalación de Moodle en una máquina con SO Mac OS X 10.5 Servidor

Esta guía de instalación no puede ser de utilidad para una instalación en MAMP o XAMPP. Si lo que se está buscando es una manera fácil de instalar Moodle en su máquina local, use [la instalación de los paquetes para Mac OS X 10.3/10.4/10.5](#) que se puede descargar de <http://download.moodle.org/macosex/>.

### 2.2.6.1 Requisitos del sistema

- Apple Computer + Apple Computer
- Mac mini or Xserve + Mini Mac o Xserve
- Processor: Intel Core 2 Duo or PPC (G4, G5) + Procesador: Core 2 Duo de Intel o PPC (G4, G5)
- RAM: 1 GB or better + RAM: 1 GB o más
- Hard Disk: 500 MB free + Disco duro: 500 MB libres
- System Software: Mac OS X 10.5 Server + Sistema de Software: Mac OS X 10.5 Servidor

### 2.2.6.2 Instalar la librería GD en el Mac OS X 10.5 Servidor

El Mac OS X Server 10.5 Leopard viene junto con PHP 5.2.4, esto es válido para instalar Moodle, pero es necesario el apoyo de GD Library.

1. Para ver qué versión de PHP está instalada en su servidor debe editar el archivo / Library / WebServer / Documentos / info.php. Se debe activar la llamada a la función `phpinfo()` mediante la supresión de las dos barras // al comienzo de la función de línea. Después de guardar el archivo puede conseguir todas las informaciones de PHP en su navegador.

```
<? php
//Se puede utilizar Server Admin para activar el módulo de Apache PHP,
se desactivada por defecto.

//Se puede comentar la directiva phpinfo()

// Pero tenga en cuenta que ésta muestra la información relativa a su
configuración del host.
phpinfo();
?>
```

Para obtener esta librería GD de apoyo al Servidor de su Mac, la manera más fácil es que descargue el archivo PHP 5.2.5 (beta6 o posterior). Inténtelo en la siguiente dirección <http://www2.entropy.ch/download/php5-5.2.5-6-beta.tar.gz> o busque en el foro [PHP en Mac OS X](#) (se encuentra en inglés).

2. Descargue y descomprima el paquete. Mueva la carpeta "php5" a /usr/local/... por lo que tendrá /usr/local/php5. Cambie la contraseña del root.

```
sudo mv /usr/local/php5 ~/Desktop/php5.old
curl -O http://www2.entropy.ch/download/php5-5.2.5-6-beta.tar.gz
tar-xzf php5-*-beta.tar.gz
sudo mv /usr/local/
sudo chown -R root:admin /usr/local/php5
```

3. Hay que decirle al servidor Web que se desea utilizar una nueva biblioteca de PHP a partir de ahora. Para ello, modifique el archivo /etc/apache2/httpd.conf.
4. Comente la línea "LoadModule php5\_module libexec/apache2/libphp5.so" añadiendo # como primer carácter de la línea.
5. Añada la nueva línea "LoadModule php5\_module local/php5/libphp5.so".
6. Guarde el archivo.

```
#LoadModule php5_module libexec/apache2/libphp5.so
LoadModule php5_module local/php5/libphp5.so
```

7. Ahora hay que modificar el archivo php.ini, éste se encuentra dentro del nuevo paquete.

Edite /usr/local/php5/lib/php.ini

Para poner en contacto la base de datos MySQL con PHP tiene que definir el valor por defecto que toma el nombre local con el que se conecta MySQL.

8. Guarde el archivo

```
mysql.default_socket = /var/mysql/mysql.sock
```

En caso de que no funcione la línea anterior, pruebe a escribir:

```
mysql.default_socket = /tmp/mysql.sock
```

9. Reinicie el servidor web.
10. Debería comprobar de nuevo los datos que muestra el servidor: <http://server-address/info.php>

Verá el número de la versión PHP 5.2.5 y el funcionamiento librería GD apoyo.



Figura 15 Usuario desarrollador: phpinfo1 en Mac

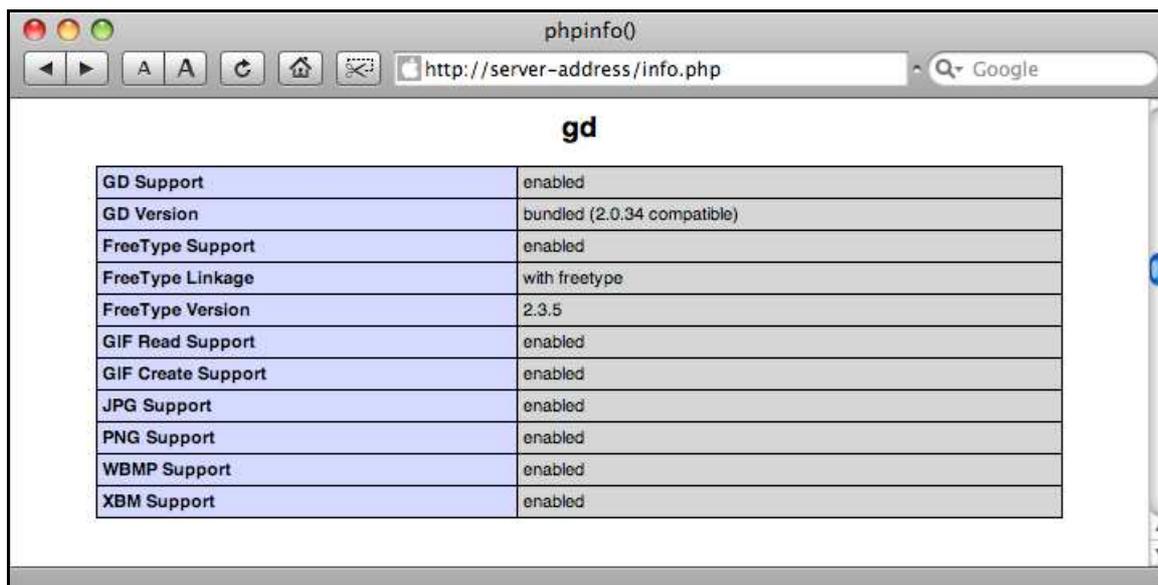


Figura 16 Usuario desarrollador: phpinfo2 en Mac

### 2.2.6.3 Configuración de MySQL

Hay que asegurarse de que la base de datos está funcionando en el Mac.

Pasos para configurarlo en modo gráfico:

1. Iniciar el administrador del servidor, para ello vaya a Aplicaciones → Server → Server Admin.
2. Activar MySQL en el servidor local.

3. Debe establecer la contraseña del usuario root antes de poder comenzar en MySQL.
4. Para la próxima vez debe descargar la base de datos MySQL herramienta de administración de phpMyAdmin <http://www.phpmyadmin.net>.
5. Mueva a la carpeta del servidor Web los documentos **/Library/ WebServer/Documents/ phpMyAdmin** y añadir la frase de seguridad para el archivo **config.inc.php**.
6. Ahora podrá utilizar <http://server-address/phpMyAdmin> en su navegador y acceder a la base de datos como el usuario *root*.
7. Añadir una base de datos *Moodle*.
8. Añadir un usuario de base de datos *Moodle* en el servidor *localhost*. No olvide poner una contraseña segura.
9. La base de datos de usuario debe ser *Moodle*, esta permite administrar la base de datos sólo *Moodle*. La instalación de phpMyAdmin es una buena solución para buscar en la base de datos mientras se está ejecutando Moodle. Además, es mucho más fácil de ver en la interfaz gráfica que en las herramientas de línea de comandos.

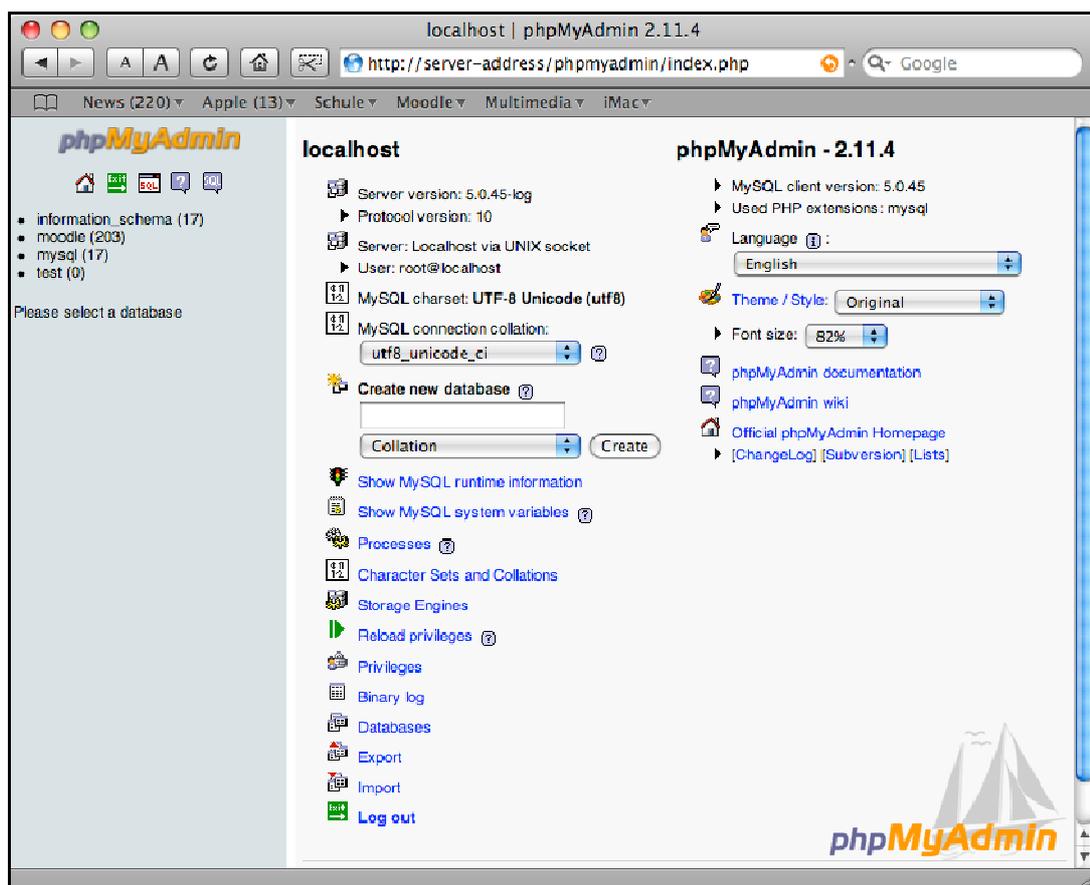


Figura 17 Usuario desarrollador: phpMyAdmin en Mac

### 2.2.6.4 Uso de línea de comandos

Pasos para hacer todo lo anterior desde línea de comandos:

1. Iniciar la línea de comandos. Lo encontrará en **Applications → Tools → Terminal**.
2. En primer lugar es necesario configurar la contraseña de root, para ello escriba:

```
Server:~ $ mysqladmin -u root password 'secret'

Server:~ $ mysqladmin -u root -p password 'topsecret'
```

3. Los próximos pasos son la creación de una nueva base de datos *de Moodle* y la creación de una base de datos de usuarios *de Moodle* para administrar la dicha base de datos.

```
Server:~ $ mysqladmin -u root -p create moodle

Server:~ $ mysql -u root -p

mysql> GRANT ALL PRIVILEGES ON moodle.* TO moodle@localhost
IDENTIFIED BY 'moodle!';

mysql> quit

Server:~$
```

### 2.2.6.5 Copiar los archivos de Moodle en el Servidor Web

Descargue Moodle de la siguiente dirección: <http://download.moodle.org> y guarde el archivo descomprimido en la carpeta /Library/WebServer/Documents/

### 2.2.6.6 Instalación de Moodle

Ya tiene en su máquina la plataforma Moodle, ahora hay que ejecutar la instalación de Moodle.

La instalación no difiere de cualquiera realizada en una máquina con SO Windows, por lo que continúe la instalación a partir del [punto 4](#) de la instalación de Moodle en una máquina con SO Windows XP.

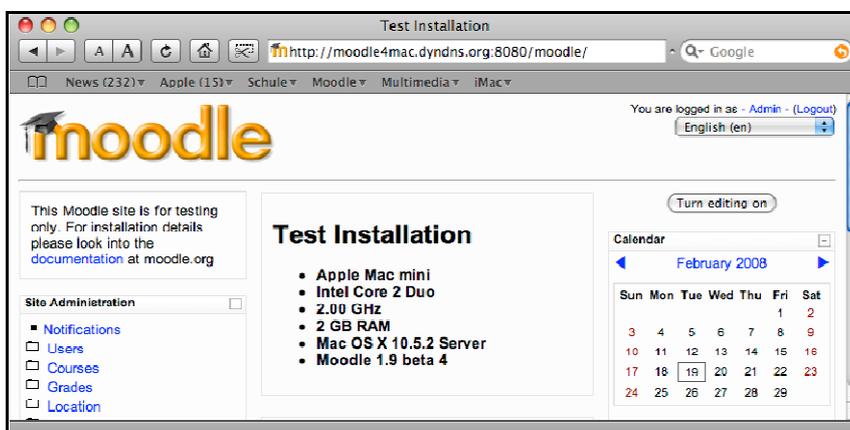


Figura 18 Usuario desarrollador: página principal del curso en mac

## 2.2.7 Instalación de Xampp en una máquina con SO Ubuntu

Para instalar el servidor de páginas Web y el servidor de bases de datos puede hacer uso del paquete XAMPP, que incluye el servidor de páginas Web Apache, el servidor de bases de datos MySQL, y phpMyAdmin, una herramienta que sirve para gestionar bases de datos MySQL y tiene una versión para instalarla en un ordenador con SO Ubuntu.

Pasos para instalar XAMPP en Ubuntu:

1. Comience bajando el paquete desde el siguiente enlace:

<http://www.apachefriends.org/de/xampp-linux.html#52>

2. A continuación, copie el paquete que ha descargado al directorio /opt, abra el terminal (Aplicaciones/Accesorios/Terminal) y descomprima el fichero:

```
cd /opt
sudo tar -zxvf xampp-linux-1.6.tar.gz
```

3. Una vez que ha hecho esto, ya tiene instalado **XAMPP**. Para iniciar cada uno de los servidores ejecute la siguiente orden:

```
sudo /opt/lampp/lampp start
```

En el terminal deben aparecer las siguientes líneas que indican que XAMPP se ha iniciado correctamente:

```
XAMPP:Starting Apache with SSL (and PHP5)...
XAMPP: Starting MySQL...
XAMPP: Starting ProFTPD...
XAMPP for Linux started...
```

4. Ahora puede probar el servidor Apache abriendo el navegador y escribiendo <http://localhost> en la barra de direcciones. Debe aparecer una página como la siguiente:





Figura 21 Usuario desarrollador: phpMyAdmin en Linux

## 2.2.8 Instalación de Moodle en una máquina con SO Ubuntu

Pasos para una correcta instalación de Moodle en su equipo:

1. Descomprima el archivo que ha bajado en el directorio /opt/lampp/htdocs/. Deberá asegurarse que después de descomprimir el archivo se crea un directorio llamado Moodle en el directorio mencionado anteriormente.
2. En su navegador escriba <http://localhost/phpmyadmin>. En la página que aparece hay un cuadro de texto, en la zona central, en el que deberá escribir el nombre de la base de datos a crear. A esta base de datos la deberá llamar **moodle**.



Figura 22 Usuario desarrollador: phpInfo en Linux

3. Cuando haya creado la base de datos escriba la siguiente dirección en el navegador para comenzar con la instalación de **Moodle**: <http://localhost/moodle>.
4. Los pasos para la instalación de Moodle en su máquina con SO Ubuntu no difieren de las de cualquier equipo con SO Windows. Continúe la instalación a partir del [punto 4.](#)



## 2.3 Manual de estilo de código

Este manual de estilo ha sido creado para conseguir que todo el código de Moodle cumpla estas características, de forma que sea más fácil de entender a los futuros desarrolladores. Es cierto que la parte más antigua del código no cumple estas normas en algunos casos, pero la comunidad Moodle está trabajando para solucionarlo gradualmente. Todo el código nuevo definitivamente deberá adherirse a estos estándares de la forma más exacta posible.

### 2.3.1 Reglas generales

Este apartado se basa en las “Reglas Generales” que propone Moodle para sus desarrolladores.

- **Todos los archivos de código deberían utilizar la extensión .php.**
- **Todas las plantillas deberían utilizar la extensión .html.**
- **Todos los archivos de texto deberían utilizar el formato de texto Unix** (la mayoría de los editores de texto tienen esto como una opción).
- **Todas las etiquetas php deben ser 'completas' como <?Php ?> ...** No 'reducidas' como <? ?>.
- **Todos los avisos de copyright deben ser mantenidos.** Puede incluir los suyos propios si resulta necesario.
- **Todos los archivos deben incluir el archivo principal config.php.**
- **Cualquier otro include/require debería utilizar una ruta absoluta que comience por \$CFG->dirroot o \$CFG->libdir,** nunca relativos, ya que estos en algunas ocasiones funcionan de forma extraña en PHP.
- **Cada archivo debería comprobar que el usuario está autenticado correctamente,** utilizando las funciones require\_login() y isadmin(), isteacher(), iscreator() o istudent().
- **Todos los accesos a la base de datos deberían utilizar las funciones definidas en lib/datalib.php** cuando sea posible - esto permite la compatibilidad con un gran número de bases de datos. Debería encontrar que prácticamente todo es posible utilizando estas funciones. Si quiere escribir código SQL entonces deberá comprobar que: funciona en cualquier plataforma; restringido a funciones específicas de su código (normalmente un archivo lib.php); y claramente comentado.
- **No cree o utilice variables globales distintas de las estándar \$CFG, \$SESSION, \$THEME, \$SITE, \$COURSE y \$USER.**
- **Todas las variables deberían ser inicializadas** o, al menos, comprobada su existencia utilizando isset() o empty() antes de ser utilizadas.

- **Todas las cadenas deberían ser traducibles** - cree nuevos textos en los archivos "lang/es\_utf8" con palabras reducidas en inglés y su traducción completa al Español y recupérelas en su código utilizando las funciones `get_string()` or `print_string()`.
- **Todos los errores deberían ser visualizados utilizando la función `print_error()`** para maximizar la traducción y ayudar a los usuarios (Automáticamente se enlaza con Moodle Docs).
- **Todos los ficheros de ayuda deben ser traducibles** - cree nuevos textos en el directorio "lang/es\_utf8/help" y llámelos utilizando la función `helpbutton()`. Si necesita actualizar un fichero de ayuda:
  - para un pequeño cambio, donde la traducción antigua del fichero podría tener todavía sentido, está permitido que haga el cambio, pero debería notificárselo a [translation@moodle.org](mailto:translation@moodle.org)
  - para un cambio importante tendrá que crear un nuevo fichero añadiéndole en el nombre un número incrementado (Ej. `filename2.html`) para que los traductores puedan ver fácilmente que se trata de una nueva versión del archivo. Obviamente el nuevo código y los índices de las páginas de ayuda deben ser modificados para apuntar a las versiones más recientes.
- **La información que llega desde el navegador** (enviada con los métodos GET o POST) **automáticamente tiene las "magic\_quotes" aplicadas** (sin importar la configuración de PHP) por lo que puede insertarla con total seguridad en la base de datos. El resto de la información (obtenida desde los archivos, o desde la base de datos) debe ser escapada con la función `addslashes()` antes de insertarla en la base de datos.
- **MUY IMPORTANTE: Todos los textos dentro de Moodle, especialmente aquellos que han sido introducidos por los usuarios, deben ser mostrados utilizando la función `format_text()`. Esto asegura que el texto es filtrado y limpiado correctamente.**
- **Las acciones de los usuarios deberían ser grabadas utilizando la función `add_to_log()`.** Estos registros son utilizados para la generación de los "Informes de Actividad" y los registros.
- **Al generar enlaces HTML, hágalos siempre relativos a la raíz del sitio Moodle**, por ejemplo, enlace a `CFG->wwwroot/mod/blonk/view.php?id=99` en lugar de únicamente `view.php?id=99`. Esto causa que su código funcionará aunque sea llamado por un script que se encuentre en otra carpeta diferente.

### 2.3.2 Estilo del código

Es necesario cumplir las siguientes normas, ya que cuando se quiere entender el diseño que sigue moodle es muy frustrante no comprender el código correctamente, ya sea por falta de estructuras o por falta de comentarios.

- El sangrado del texto debe ser siempre de 4 espacios. **No utilice los tabuladores NUNCA.**

- Los **nombres de las variables** tienen que ser siempre fáciles de leer, procurando que sean palabras en **minúsculas con significado en Inglés**. Si realmente necesita más de una palabra, póngalas juntas, pero procure mantenerlas tan breves como sea posible. Utilice nombres en plural para las matrices de objetos.

```
BIEN: $quiz
BIEN: $errorstring
BIEN: $assignments (for an array of objects)
BIEN: $i (but only in little loops)
MAL: $Quiz
MAL: $aReallyLongVariableNameWithoutAGoodReason
MAL: $error_string
```

- Las **constantes** tienen que definirse **siempre en mayúsculas, y empezar siempre por el nombre del módulo al que pertenecen**. Deberían tener las palabras separadas por guiones bajos.

```
define("FORUM_MODE_FLATOLDEST", 1);
```

- Los **nombres de las funciones** tienen que **ser palabras sencillas en minúsculas y en Inglés, y empezar con el nombre del módulo al que pertenecen** para evitar conflictos entre módulos. Las palabras deberían **separarse por guiones bajos**. Los parámetros, si es posible, tendrán valores por defecto. Compruebe que no haya espacio entre el nombre de la función y los paréntesis.

```
function forum_set_display_mode($mode=0) {
    global $USER, $CFG;

    if ($mode) {
        $USER->mode = $mode;
    } else if (empty($USER->mode)) {
        $USER->mode = $CFG->forum_displaymode;
    }
}
```

- Los **bloques de código siempre deben estar encerrados por llaves** (incluso si solo constan de una línea). Moodle utiliza el siguiente estilo:

```
if ($quiz->attempts) {
    if ($numattempts > $quiz->attempts) {
        error($strtoomanyattempts, "view.php?id=$cm->id");
    }
}
```

- Las **cadenas tienen que ser definidas utilizando comillas simples siempre que sea posible**. Esto sirve para obtener un mejor rendimiento.

```
$var = 'some text without any variables';
$var = "with special characters like a new line \n";
$var = 'a very, very long string with a '.$single.' variable in it';    $var = "some
$text with $many variables $within it";
```

- Los **comentarios deben ser añadidos de forma que resulten prácticos**, para explicar el flujo del código y el propósito de las funciones y variables.
- **Cada función (y cada clase) debería utilizar el popular formato phpDoc**. Esto permite que la documentación sea generada automáticamente.
- Los **comentarios en línea** deberían utilizar los caracteres **//**, alineados con cuidado por encima de las líneas de código que comenta.

```
/**
 * The description should be first, with asterisks laid out exactly
 * like this example. If you want to refer to a another function,
 * do it like this: {@link clean_param()}. Then, add descriptions
 * for each parameter as follows.
 *
 * @param int $postid The PHP type is followed by the variable name
 * @param array $scale The PHP type is followed by the variable name
 * @param array $ratings The PHP type is followed by the variable name
 * @return mixed
 */
function forum_get_ratings_mean($postid, $scale, $ratings=NULL) {
    if (!$ratings) {
        $ratings = array(); // Initialize the empty array
        if ($rates = get_records("forum_ratings", "post", $postid)) {
            // Process each rating in turn
            foreach ($rates as $rate) {
                ....etc
            }
        }
    }
}
```

- **El espacio en blanco se puede utilizar con bastante libertad** - no se preocupe por separar las cosas un poco para ganar en claridad. Generalmente, debería haber un espacio entre llaves y líneas normales y ninguno entre llaves y variables o funciones:

```
foreach ($objects as $key => $thing) {
    process($thing);
}

if ($x == $y) {
    $a = $b;
} else if ($x == $z) {
    $a = $c;
} else {
    $a = $d;
}
```

- Cuando esté realizando una **COPIA de un objeto**, utilice siempre la función **clone()** originalmente sólo disponible en php5 (en caso contrario simplemente tendrá una referencia al primer objeto). Moodle le garantiza que este método funcionará también bajo php4.

MAL:	\$b = \$a;
BIEN:	\$b = clone(\$a);

- Si lo que se quiere **copiar no es un objeto, pero puede contener objetos** (Ej. un array de objetos) utilice la función **fullclone()** en su lugar

### 2.3.3 Estructura de la base de datos

- **Cada tabla** debe tener un campo autonumérico **id** (INT10) **como clave primaria**.
- La **tabla principal** que contiene instancias de cada módulo debe tener el mismo nombre que el módulo y contener, por lo menos, los siguientes campos:
  - **id** - clave primaria
  - **course** - el identificador del curso al que la instancia pertenece
  - **name** - el nombre completo de la instancia
- El **resto de las tablas** asociadas con un módulo que contiene información sobre 'cosas', deberían ser llamadas **modulo\_cosas** (fíjese en el plural).

- Los nombres de las tablas y de los campos tienen que **evitar el uso de palabras reservadas por las Bases de Datos**. Por favor, compruébelo antes de crearlas.
- Los **nombres de los campos** (columnas) **deberían ser sencillos y cortos**, siguiendo las mismas reglas que los nombres de las variables.
- Cuando sea posible, **las columnas que contengan una referencia al campo id de otra tabla** (por ejemplo, modulo) **debería ser llamado moduloid**. (fíjese que esta norma es nueva y no es seguida por algunas tablas antiguas).
- **Los campos booleanos serán implementados como enteros cortos** (por ejemplo, INT4) con los valores 0 o 1, para permitir la futura expansión de los valores si fuera necesario.
- La mayoría de las tablas tienen que tener un campo **timemodified** (INT10) que será actualizado con la fecha actual (timestamp de UNIX) obtenida con la función time() de PHP.
- **Defina siempre un valor por defecto para cada campo** (y haga que tenga sentido).
- **Cada tabla debe comenzar con el prefijo de la base de datos** (\$CFG->prefix). En muchos casos esto es gestionado automáticamente. Además, bajo PostgreSQL, el nombre de cada índice debe empezar también con el prefijo.
- **Para garantizar la compatibilidad entre bases de datos**, por favor, siga las reglas siguientes sobre el uso del comando **AS** (solo si necesita alias en tablas/campos, por supuesto):
  - No utilice el comando AS para alias de tablas.
  - Utilice el comando AS para alias de campos (columnas).
- **Nunca cree UNIQUE KEYS (restricciones) para nada. En su lugar utilice UNIQUE INDEXes.** En el futuro, si se decide añadir integridad referencial a Moodle y si se necesitan UNIQUE KEYS, serán utilizadas, pero no por ahora. Por favor, fíjese que el Editor XMLDB permite especificar tanto restricciones UNIQUE y FOREIGN (y eso es bueno, teniendo el XML bien definido), pero solo los índices subyacentes serán realmente generados en la DB.
- El uso de UNIQUE KEYS creadas en el Editor XMLDB (lea el punto anterior) solo debe ser definida si el campo/campos van a ser el objetivo para alguna FOREIGN KEY (a nivel de Editor). En caso contrario, créelas como UNIQUE INDEXes.
- **Las tablas asociadas con un bloque deben seguir las siguientes convenciones en sus nombres:** \$CFG->prefix + "block\_" + nombre del bloque + añadidos. Por ejemplo, asumiendo que \$CFG->prefix es 'mdl\_', todas las tablas para el bloque "rss\_client" deberán empezar por 'mdl\_block\_rss\_client' (siendo posible añadir más palabras al final, Ej. 'mdl\_block\_rss\_client\_anothertable'...). Esta regla será completamente forzada con Moodle 2.0, dando algo de tiempo a los desarrolladores hasta entonces.

- **Nunca realice cambios a la base de datos en ramas estables.** Si hace eso, entonces los sitios actualizando de una versión estable a la siguiente pueden encontrarse con cambios por duplicado, lo cual puede producir errores serios.
- **Cuando haga referencia a una variable entera en consultas SQL, no entrecomille el valor.** Por ejemplo, `get_records_select('question', "category=$catid")` es correcto. `get_records_select('question', "category='$catid'")` es incorrecto. Ese uso oculta posibles errores cuando `$catid` está sin definir. (Esta discusión lo explica.)

### 2.3.4 Normas de seguridad (y control de la información de formularios y URLs)

- **No se base en 'register\_globals'.** Cada variable debe ser correctamente inicializada en cada fichero de código. Debe ser obvia la procedencia de cada variable.
- **Inicialice todos los arrays y objetos aunque estén vacíos.** `$a = array()` o `$obj = new stdClass();`.
- **No utilice la función `optional_variable()`.** En su lugar, utilice la función `optional_param()`. Seleccione la opción `PARAM_XXXX` apropiada al tipo de parámetro que espera. Para comprobar y definir un valor opcional para una variable, utilice la función `set_default()`.
- **No utilice la función `require_variable()`.** En su lugar, utilice la función `required_param()`. Seleccione la opción `PARAM_XXXX` apropiada al tipo de parámetro que espera.
- **Utilice `data_submitted()`, con cuidado.** La información todavía debe ser limpiada antes de utilizarla.
- **No utilice `$_GET`, `$_POST` o `$_REQUEST`.** En su lugar, utilice las funciones `required_param()` o `optional_param()` apropiadas.
- **No compruebe las acciones con código como: `if (isset($_GET['algo']))`.** Utilice, por ejemplo, `$algo = optional_param( algo, -1, PARAM_INT )` y entonces compruebe que está dentro de los valores esperados, por ejemplo, `if ($something >= 0) {...`
- **Cuando sea posible agrupe todas sus llamadas a `required_param()`, `optional_param()` y el resto de inicialización de variables en el principio de cada fichero** (o función) para que sea fácilmente localizable.
- **Utilice el mecanismo 'sesskey' para proteger el envío de formularios de ataques.** Un ejemplo de uso: cuando el formulario es generado, incluya `<input type="hidden" name="sesskey" value="<?php echo sesskey(); ?>" />`. Cuando el formulario es procesado, compruebe `if (!confirm_sesskey()) {error('Bad Session Key');}`.
- **Todos los nombres de ficheros deben ser 'limpiados' utilizando la función `clean_filename()`,** si esto no ha sido realizado con el uso de las funciones `required_param()` o `optional_param()` con anterioridad.

- **Cualquier información leída desde la base de datos debe tener la función addslashes()** aplicada antes de volver a enviar la información a la base de datos. Un objeto completo puede ser procesado con la función addslashes\_object().
- **No utilice información obtenida de \$\_SERVER si puede evitarlo.** Presenta algunos problemas de portabilidad.
- Si no ha sido realizado en ningún otro lugar, **asegúrese de que la información enviada a la base de datos ha sido filtrada** mediante la función clean\_param() utilizando la opción PARAM\_XXXX apropiada.
- **Si escribe código SQL, asegúrese completamente de que es correcto.** En particular, compruebe la falta de comillas en las variables utilizadas. Es un punto de entrada para ataques de tipo 'SQL injection'.
- **Compruebe toda la información** (especialmente la que es enviada a la base de datos) en cada archivo que es utilizada. Nunca confíe en que otro código estará haciendo ese trabajo.
- **Los bloques de código que se incluyan deben presentar una estructura PHP correcta** (por ejemplo, con una declaración de una clase, de funciones, etc.) - los bloques de código lineales ("espagueti") suelen tender a utilizar variables sin inicializar (y son menos legibles).
- **Si necesita usar shell\_exec()** (o cualquier otra función que invoque un shell), **asegúrese de que ha limpiado los parámetros anteriormente** con escapeshellcmd()/escapeshellarg() (de lo contrario abrimos la puerta a ataques de inyección de shell).

## 2.4 Manual de estilo de interfaz

Este documento todavía está en construcción, por lo que pueden aparecer nuevos apartados continuamente.

### 1. Simplicidad

Use el mínimo de interfaz necesario para obtener un trabajo terminado.

### 2. Páginas estándar

- Módulos de actividad
  - ***index.php*** - lista todas las opciones de ese módulo en un curso.
  - ***view.php*** - muestra una opción en particular.
  - ***config.html*** - configura una opción del módulo.

### • Bloques

- ***config.html*** - configura una opción del bloque.

### 3. Un script por función o página principal

- Plantilla de página
- Imprima las cabeceras con `print_heading`, use los *hooks* CSS para IDs y Clases.
- Imprima las cajas alrededor del texto usando `print_simple_box`, use los *hooks* CSS para IDs y Clases.

### 4. Plantilla de formulario

- Muestre las opciones más importantes en la parte superior.
- Cada entrada debe tener una etiqueta y, si es necesario, un archivo de ayuda.
- Si hay más de 10 opciones, desglóselas en los parámetros necesarios y opcionales, extra o avanzados.

### 5. Manejo de tablas

- Use la función `print_table` cuando sea posible.

### 6. Herramientas de navegación estándar

- Todas las páginas deberían llamar a `print_header`, y suministrar una ruta de navegación estándar que apareciera allí. Donde sea posible, debería verse como: COURSE >> INDEX >> INSTANCE >> SUBPAGES...

- Las páginas incluidas en módulos de actividad deberían llamar a `navmenu()` para generar el menú de navegación apropiado.

#### 7. Direcciones URL

- Las direcciones URL deben ser lo más cortas posible.
- No usar subrayado en nombres de parámetros o nombres de archivos.
- Nunca use dos palabras cuando una sea suficiente.

#### 8. Botones o enlaces

El *Web Accelerator* de Google proporciona algunas sugerencias:

- Las acciones que puedan modificar el estado de Moodle (archivos de datos, base de datos, información de sesión) deben ser llevadas a cabo por medio de botones.
- Como mínimo, tales acciones -que son implementadas como enlaces- deben remitir a una página de confirmación que sí utilice botones.

#### 9. Enlaces a archivos de ayuda

- Los botones de ayuda deben estar a la derecha del objeto (excepcionalmente pueden estar a la izquierda, si el objeto está alineado a la derecha)



## 2.5 CVS para desarrolladores

CVS es un Sistema Concurrente de Versiones, una forma de trabajo habitualmente utilizada para almacenar el código fuente de grandes proyectos de software. CVS almacena todas las versiones de todos los ficheros de tal forma que nada es nunca perdido, y su utilización por varias personas es registrada. También proporciona una forma de combinar el código de dos o más personas que estén trabajando simultáneamente en el mismo fichero. Todo el código y sus versiones son almacenados en un servidor central (en el caso de Moodle, el de Sourceforge). Puede encontrar una completa documentación del sistema CVS en el libro online "Open Source Development with CVS".

### 2.5.1 Unirse al proyecto como desarrollador

Para poder escribir cambios en el archivo de CVS Moodle, primero necesita tener **una cuenta en Sourceforge** (el registro es libre y sencillo). Para los ejemplos en esta página asumiremos que su nombre de usuario es *miusuario* y su contraseña es *micontraseña*. Preste especial atención a las instrucciones de sourceforge para crear su directorio de CVS - algo que tendrá que hacer con cada cuenta para "activarla" para CVS. Básicamente tiene que utilizar ssh para conectarse interactivamente con cvs.sourceforge.net.

**Una vez que tenga una cuenta funcionando** en Sourceforge, **contacte con Martin Dougiamas** (creador y director de proyecto de todo el sistema Moodle) para que él pueda configurar su cuenta con los accesos correspondientes a los directorios de Moodle.

**Para evitar que cada vez que ejecute un comando de CVS el sistema le pregunte micontraseña, siga las instrucciones de Sourceforge para utilizar claves autorizadas.** Este paso es opcional, pero puede hacer su experiencia con CVS mucho más cómoda.

Con todo esto hecho, debería tener todos los permisos necesarios, así que únicamente **necesita configurar su ordenador y descargar los fuentes actuales para empezar a trabajar con ellos.** No es necesario leer la documentación de Sourceforge CVS documentation (inglés) a no ser que esté interesado.

### 2.5.2 Módulos CVS

Dentro de CVS la palabra "módulos" hace referencia a colecciones separadas de código. En Moodle tenemos los siguientes módulos en nuestro repositorio:

- **moodle** - el código fuente principal de Moodle
- **contrib** - contribuciones de los usuarios y todo tipo de código en desarrollo
- **mysql** - un phpMyAdmin personalizado para trabajar con la base de datos de Moodle

- **windows-cron** - un pequeño paquete que hace posible el funcionamiento del cron en Windows
- **docs** - variada documentación extra generada por los usuarios

La mayoría de la gente está trabajando en las funcionalidades existentes en el módulo moodle, pero muchos también están contribuyendo con nuevas ideas en el módulo contrib. Una vez que el código alcanza cierto nivel de madurez en la zona de contrib puede ser movido a la zona principal de moodle.

## 2.5.3 Comandos básicos

### 2.5.3.1 CVS en UNIX

El sistema CVS de Sourceforge utiliza ssh como una capa de transporte para seguridad, por lo que tendrá que tener definida la variable de entorno CVS\_RSH en su sesión de Unix. Lo mejor es poner estos comandos en sus archivos .bashrc o .cshrc para que no tenga que teclearlos repetidamente:

```
setenv CVS_RSH ssh (para csh, tcsh etc)
export CVS_RSH=ssh (para sh, bash etc)
```

A continuación puede obtener (checkout) la última versión de desarrollo de Moodle usando la siguiente línea:

```
cvs -z3 -d:ext:miusuario@moodle.cvs.sourceforge.net:/cvsroot/moodle co moodle
```

NOTA: No intente ejecutar su primer comando CVS sobre una instalación de Moodle: empiece de cero en un directorio vacío.

El comando es similar para otros módulos CVS:

```
cvs -z3 -d:ext:miusuario@moodle.cvs.sourceforge.net:/cvsroot/moodle co contrib
```

Cada vez que ejecute un comando CVS le te preguntará *micontraseña* a no ser que configure las claves autorizadas.

Ahora, debería tener un nuevo directorio 'moodle'. Puede renombrarlo y moverlo donde quiera. Entre en él:

```
cd moodle
```

Todos los ficheros de Moodle deberían estar ahí. Ahora puede modificarlos localmente. Para comparar sus ficheros y directorios con los que existen en la copia principal del servidor CVS utilice cvs diff, Ej.:

```
cvs diff -c config-dist.php
cvs diff -c lang
```

Para obtener las últimas actualizaciones desde el servidor utilice:

```
cvs update -dP
```

Para copiar sus nuevos ficheros de vuelta al servidor debería hacer algo como:

```
cd lang/ca
cvs commit
```

Se le requerirá que añada algunos comentarios sobre los cambios (depende de su editor de texto por defecto) ... escriba algún comentario con significado y cierre el editor ... los ficheros serán enviados a Sourceforge y almacenados allí.

Para ahorrar algo de tiempo, puede poner algunos argumentos por defecto en un fichero llamado .cvsrc en su directorio. En nuestro ejemplo contiene:

```
diff -c
update -dP
```

Utilice 'cvs help' para obtener más información...

### 2.5.3.2 CVS en Windows

En primer lugar necesita descargar una copia nueva de Moodle utilizando su cuenta de desarrollador:

1. Descargue TortoiseCVS desde tortoisecvs.org e instálelo, entonces reinicie.
2. Busque o cree una carpeta en la que quiera que Moodle sea descargado.
3. Selecciónela con el botón derecho del ratón y seleccione "CVS Checkout" en el menú. Verás una caja de diálogo.
4. Copie este texto en el campo CVSROOT (¡utilizando su propio nombre de usuario!):

```
:ext:miusuario@moodle.cvs.sourceforge.net:/cvsroot/moodle
```

5. En el campo "Module", teclee "moodle" para obtener la última versión de Moodle, "contrib" para obtener el directorio de contribuciones, o "mysql" para descargar el módulo MySQL Admin.
6. Seleccione el botón "OK" y todo debería ser descargado.

Un diálogo debería mostrarle todos los ficheros que están siendo descargados y, tras un rato, debería tener una copia completa de Moodle. Después de esta primera descarga (checkout) puede obtener los últimos cambios desde el servidor CVS:

7. Con el botón derecho del ratón encima de su carpeta Moodle (o de cualquier otro fichero) seleccione "CVS Update".
8. Examine los registros atentamente. Apunte los conflictos que pudieran aparecer si ha realizado modificaciones en su copia local que entran en conflicto con cambios en las versiones que le este descargando - necesitará editar estos ficheros y resolver los conflictos manualmente.

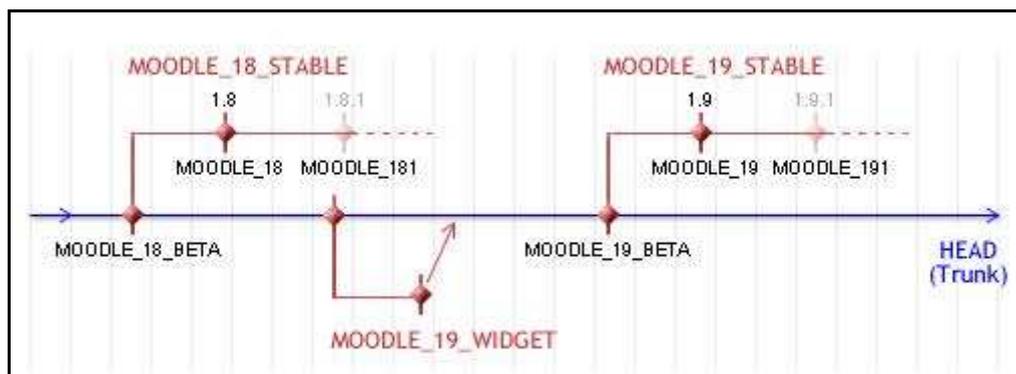
Después de modificar ficheros (notará que sus iconos cambian de verde a rojo) puede enviar esos cambios al servidor CVS así:

9. Con el botón derecho del ratón encima de su carpeta Moodle (o de cualquier otro fichero) seleccione "CVS Commit...".

10. En el diálogo que aparece, introduzca una descripción clara de los cambios que está enviando.
11. Seleccione "OK". Sus cambios serán enviados al servidor.

### 2.5.4 Trabajando con ramas

Este diagrama muestra como el módulo moodle se divide en diferentes ramas (branches) a lo largo del tiempo.



*Figura 23 Usuario desarrollador: ramas de Moodle*

Para ver todas las etiquetas y ramas existentes que están disponibles, utilice este comando en cualquier fichero antiguo (como el index.php en el directorio principal de Moodle):

```
cvstatus -v index.php
```

Algunas instrucciones sobre las etiquetas (tags):

1. Los nombres de las etiquetas y de las ramas siempre deben ser escritos en mayúsculas.
2. **Las etiquetas y las ramas deben ser SIEMPRE aplicadas a todo el módulo** (todo Moodle). No cree etiquetas para ficheros o carpetas individuales.
3. **No se permite renombrar etiquetas** porque muchas personas las utilizan, así que ¡créelas bien la primera vez!

#### 2.5.4.1 Desarrollando en el tronco (trunk)

El Tronco del CVS es la versión de desarrollo principal de Moodle. En CVS también es conocido como el HEAD, o rama por defecto (default branch).

Los desarrolladores de Moodle intentan mantenerlo tan estable como es posible, pero normalmente contiene nuevo código con errores y pequeñas inestabilidades.

En cualquier momento se puede decidir que el producto tiene las funcionalidades necesarias para publicar una nueva versión. En ese momento, el tronco es marcado con una etiqueta

MOODLE\_XX\_BETA (por si en algún momento se quiere volver a ese punto) y una nueva rama es creada para esa versión con el nombre MOODLE\_XX\_STABLE.

En ese momento, un paquete Beta es también publicado - es para los usuarios que no utilizan el CVS pero quieren comprobar las nuevas funcionalidades y ayudar en la detección de errores.

### 2.5.4.2 Ramas (branches) estables para cada versión

En cuanto la rama estable MOODLE\_XX\_STABLE es creada, los esfuerzos de los desarrolladores se dividirán en dos objetivos durante cierto tiempo. Algunas personas continuarán trabajando en nuevas funcionalidades en el tronco para la próxima versión, pero la mayoría de los desarrolladores se concentrarán en utilizar la rama STABLE y solucionar los problemas que sean encontrados en ella.

Puede cambiar su copia local de Moodle a la rama STABLE utilizando el siguiente comando en Unix desde tu directorio base de Moodle:

```
cvs update -dP -r MOODLE_XX_STABLE
```

Después de esto, todos los comandos descritos anteriormente serán aplicados a la rama estable. Para volver al tronco, simplemente teclee:

```
cvs update -dPA
```

En clientes Windows debería tener un menú en el que puede seleccionar la rama que desee.

Una vez que la rama STABLE realmente se estabiliza, una versión oficial puede ser creada. Se crean los paquetes necesarios para la distribución y la rama es marcada (por Martin Dougiamas) con una etiqueta llamada MOODLE\_XXX

Periódicamente, los problemas solucionados en la rama STABLE deben ser combinados con el tronco para que también sean solucionados para futuras versiones de Moodle. Una etiqueta flotante llamada MOODLE\_XX\_MERGED debe ser mantenida para marcar el punto en el que se realizó la última combinación. El procedimiento para realizar dicha combinación es el siguiente:

1. Obtenga la última versión del tronco:

```
cvs update -dPA
```

2. Combine todo desde la rama desde la última combinación, en el tronco:

```
cvs update -kk -j MOODLE_XX_MERGED -j MOODLE_XX_STABLE
```

3. Observe atentamente los registros para ver si se producen conflictos, y solucione cada fichero que vea que presenta problemas.

4. Guarde (commit) los cambios de la combinación en el tronco CVS:

```
cvs commit
```

5. Vuelva a la rama:

```
cvs update -dPr MOODLE_XX_STABLE
```

6. Actualice la etiqueta flotante para que todo este proceso pueda desarrollarse la próxima vez:

```
cvs tag -RF MOODLE_XX_MERGED
```

7. Por último, los valores de la variable \$version en todos los ficheros version.php de Moodle en la ramas NO deben ser actualizados en la medida de lo posible (excepto el último dígito si es verdaderamente necesario). La razón para esto es que alguien actualizándose desde una versión estable a la siguiente podría perder actualizaciones de la base de datos que hayan sucedido en el tronco.

### ***2.5.4.3 Ramas de funcionalidad para grandes cambios***

De vez en cuando, puede haber alguna funcionalidad que necesita estar controlada para que distintas personas puedan trabajar en ella, pero que es muy inestable para ser incluida en el tronco de desarrollo.

En estos casos, una rama temporal es creada para trabajar en la funcionalidad y, tan pronto como sea posible, combinarla con el tronco de nuevo. En el diagrama mostrado anteriormente se ha incluido un ejemplo llamado MOODLE\_14\_WIDGET de este tipo de rama.

Si necesita hacer esto para su nueva WIDGET, siga estos pasos:

1. Discúptalo con otros desarrolladores para estar seguro de que es realmente necesaria
2. Cree una nueva etiqueta en el tronco (para todo Moodle) llamada MOODLE\_XX\_WIDGET\_PRE

```
cvs tag -R MOODLE_XX_WIDGET_PRE
```

3. Cree su rama llamada MOODLE\_XX\_WIDGET

```
cvs tag -Rb MOODLE_XX_WIDGET
```

4. Trabaje en esa rama hasta que la funcionalidad sea razonablemente estable. Envíe los cambios al servidor CVS cuando estime oportuno:

```
cvs commit
```

5. Cuando la funcionalidad esté lista, combine toda la rama en el tronco, solucione los posibles conflictos, guárdela en el tronco y abandone la rama:

```
cvs update -dPA  
cvs update -kk -j MOODLE_XX_WIDGET  
cvs commit
```

## 2.6 Desarrollo de un módulo de actividades en Moodle

Quizá lo más complicado en el desarrollo de módulos de actividades sean los primeros pasos, debido a la escasa documentación existente, especialmente en español.

Desde este apartado del documento se pretenden describir unos sencillos pasos que despejen las dudas que puedan surgir a la hora de construir un módulo, si bien ésta es sólo una guía de iniciación, el desarrollador deberá consultar la documentación existente en el propio código fuente de Moodle.

### 2.6.1 Estructura de un módulo en Moodle

Un módulo de actividades viene definido por un conjunto de ficheros php y html que se incluyen dentro de una carpeta con el nombre del módulo, que a su vez se incluye en la carpeta /mod del directorio base de Moodle. No resulta difícil encontrar estos ficheros, en forma de plantilla, en la red, aunque con escasa documentación y siempre en inglés.

Estos documentos presentan una estructura flexible, con unos campos básicos, que deben ampliarse para dar funcionalidad al módulo.

Para el funcionamiento de uno de estos módulos, es necesario que se definan, al menos, los siguientes archivos, con el contenido mínimo que se describirá a continuación:

- /mod
  - /db
    - access.php
    - install.xml
    - upgrade.php
  - /lang
    - /es\_utf8
      - /help
        - Fichero de ayuda 1.html
        - ...
        - Fichero de ayuda N.html
  - nombremodulo.php

- icon.gif
- index.php
- lib.php
- mod.html
- version.php
- view.php

Como se comentó en la guía de estilo, todos los archivos php deben incluir el archivo config.php, ubicado en el directorio base de Moodle.

### 2.6.2 Librerías de Moodle para el desarrollo de módulos

Moodle no solo facilita un entorno para la integración de nuevos módulos desarrollados en php, sino que incorpora una serie de librerías enfocadas a facilitar la labor de los desarrolladores, si bien es verdad que resulta difícil iniciarse en ellas y conocer el cometido de cada una, a pesar de encontrarse documentación en el propio código fuente.

En este apartado se enumerarán algunas de las librerías más importantes, así como las funciones básicas que todo desarrollador debería conocer antes de comenzar a construir un módulo.

Aunque con estas funciones puede comenzarse la construcción de un módulo de actividad, se recomienda recurrir a la documentación sita en el propio código fuente para desarrollar un módulo aprovechando el cien por cien de la funcionalidad de Moodle. Dichas librerías pueden encontrarse dentro del directorio /lib de Moodle. Por lo tanto, el propósito de este apartado del documento es más bien servir como un conjunto de referencias para que el desarrollador novel sepa por donde debe empezar su trabajo.

Algunas de estas librerías fueron desarrolladas por el propio equipo de Moodle, otras por terceros y, hoy en día, se distribuyen con Moodle gracias a sus licencias de código abierto.

**Tabla 6 Usuario desarrollador: listado y descripción de librerías y funciones básicas**

LISTADO Y DESCRIPCIÓN DE LIBRERÍAS Y FUNCIONES BÁSICAS	
accesslib.php	Describe funciones para el soporte de capacidades
<pre>require_capability(\$capability,     \$context=NULL,     \$userid=NULL,    \$doanything=true,     \$errormessage='nopermissions',     \$stringfile=")</pre>	<p>Comprueba que el usuario tiene cierta capacidad, si no la tiene, la página deja de ejecutarse mostrando un mensaje de error estándar.</p> <p>Parámetros:</p> <p><i>\$capability</i>: Capacidad que deseamos comprobar.</p> <p><i>\$context</i>: Contexto en el que se debe tener esta capacidad.</p> <p><i>\$userid</i>: ID del usuario cuya capacidad queremos controlar.</p> <p><i>\$doanything</i>: Establecer a false si no queremos que ocurra nada.</p> <p><i>\$errormessage</i>: Cada con el mensaje de error.</p> <p><i>\$stringfile</i>: Archivo en el que buscar \$errormessage.</p>
<pre>has_capability(\$capability,     \$context=NULL,     \$userid=NULL,     \$doanything=true)</pre>	<p>Devolverá true si un usuario <i>\$userid</i>, por defecto el usuario que esté accediendo al módulo, tiene la capacidad <i>\$capability</i> en el contexto <i>\$context</i>.</p> <p>Parámetros:</p> <p><i>\$capability</i>: Capacidad que deseamos comprobar.</p> <p><i>\$context</i>: Contexto de la capacidad.</p> <p><i>\$userid</i>: ID del usuario cuya capacidad deseamos controlar.</p> <p><i>\$doanything</i>: Si se establece a falso, la función no opera.</p>
<pre>get_context_instance(     \$contextlevel=NULL,     \$instance=SITEID)</pre>	<p>Devuelve el contexto de una instancia como un objeto.</p> <p>Parámetros:</p> <p><i>\$contextlevel</i>: Nivel del contexto, acepta los</p>

	<p>siguientes valores:</p> <p>CONTEXT_SYSTEM</p> <p>CONTEX_PERSONAL</p> <p>CONTEXT_USER</p> <p>CONTEXT_COURSECAT</p> <p>CONTEXT_COURSE</p> <p>CONTEXT_GROUP</p> <p>CONTEXT_MODULE</p> <p>CONTEXT_BLOCK</p> <p><i>\$instance</i>: Identificación de la instancia del contexto.</p>
<p>Estas tres funciones sirven para ilustrar el tipo general de las funciones de la librería, no obstante se recomienda consultar su extensa documentación para obtener información sobre otras funciones que dan soporte al sistema de capacidades.</p>	
<p>datalib.php</p>	<p>Funciones para el manejo de la información de la base de datos</p>
<pre>function get_users(\$get=true, \$search="", \$confirmed=false, \$exceptions="", \$sort='firstname          ASC', \$firstinitial="", \$lastinitial="", \$page="", \$recordsperpage="", \$fields='*')</pre>	<p>Obtiene un subconjunto de los usuarios del sitio.</p> <p>Parámetros:</p> <p><i>\$get</i>: Si se establece a falso, solo retornará una cuenta del número de registros coincidentes.</p> <p><i>\$search</i>: Una determinada cadena a buscar.</p> <p><i>\$confirmed</i>: Si se establece a true, solo devolverá los usuarios que hayan sido confirmados en el curso.</p> <p><i>\$exceptions</i>: Una lista en forma de array conteniendo ciertos identificadores de usuario a ignorar.</p> <p><i>\$sort</i>: Fragmento SQL para la ordenación de los registros a devolver.</p> <p><i>\$firstinitial</i>: Inicial del primer nombre de los usuarios a buscar.</p>

	<p><i>\$lastinicial</i>: Inicial del apellido de los usuarios a buscar.</p> <p><i>\$recordsperpage</i>: Número de registros a obtener en cada llamada, para evitar saturación.</p> <p><i>\$fields</i>: Campos de los registros a obtener (igual que el contenido de una sentencia SELECT).</p>
<p>Con un funcionamiento similar a la función <code>get_users()</code>, <code>datalib.php</code> implementa funciones para obtener los cursos del sitio o de un determinado usuario, los módulos de un determinado curso, los usuarios que han sido confirmados en el sistema, los distintos grupos establecidos, los usuarios que conforman cada grupo y más información similar, todas ellas documentadas en el archivo <code>datalib.php</code>.</p>	
<code>ddllib.php</code>	Funciones para el manejo de la estructura de la base de datos
<p>Esta librería proporciona funcionalidades avanzadas para hacer la manipulación de la estructura de la base de datos lo más portable posible, debiendo modificarse dichas estructuras usando funciones de esta biblioteca y nunca con sentencias propias de determinada base de datos.</p> <p>Se proporcionan funciones como las siguientes, cuyo nombre, en ocasiones, recuerda a las sentencias SQL habituales:</p> <p><code>create_table()</code> para crear una tabla.</p> <p><code>drop_table()</code> para borrar el contenido de una tabla.</p> <p><code>rename_table()</code> para cambiar el nombre a una tabla.</p> <p><code>add_field()</code> para añadir un campo a una tabla.</p> <p><code>change_field_tipe()</code> para cambiar el tipo de un campo.</p> <p><code>add_key()</code> para añadir una clave.</p> <p><code>add_index()</code> para añadir un índice.</p> <p>...</p>	
<code>dmllib.php</code>	Funciones para obtener, borrar, actualizar, contar e insertar registros en la base de datos.
<code>execute_sql(\$command, \$feedback=true)</code>	<p>Ejecuta una sentencia SQL e imprime el resultado si se especifica.</p> <p>Parámetros:</p>

	<p><i>\$command</i>: Sentencia SQL completa a ejecutar.</p> <p><i>\$feedback</i>: Si se establece a true, el sistema imprimirá el resultado de la ejecución.</p> <p><i>Nota sobre la función</i>: Para acceder a las tablas de la base de datos debemos precederlas del prefijo de todas las tablas de Moodle, esto se consigue introduciendo el texto “{\$CFG-&gt;prefix}nombre_tabla” (sin comillas).</p>
Funciones para comprobar la existencia de registros	
<p>record_exists(\$table, \$field1="", \$value1="", \$field2="", \$value2="", \$field3="", \$value3="")</p>	<p>Devolverá true si existe en la tabla \$table al menos un registro cuyos valores en los campos \$fieldx se correspondan con \$valuex.</p> <p>Parámetros:</p> <p><i>\$table</i>: Tabla en la que buscar los registros.</p> <p><i>\$field1</i>: Primer campo sobre el que comparar.</p> <p><i>\$value1</i>: Primer valor que comparar.</p> <p><i>\$field2</i>: Segundo campo sobre el que comparar.</p> <p><i>\$value2</i>: Segundo valor que comparar.</p> <p><i>\$field3</i>: Tercer campo sobre el que comparar.</p> <p><i>\$value3</i>: Tercer valor que comparar.</p>
<p>record_exists_select(\$table, \$select="")</p>	<p>Devolverá true si existen en la tabla \$table cuyos valores que se correspondan con el fragmento WHERE pasado como parámetro.</p> <p>Parámetros:</p> <p><i>\$table</i>: Tabla en la que buscar los registros.</p> <p><i>\$select</i>: Fragmento completo WHERE.</p>
<p>record_exists_sql(\$sql)</p>	<p>Devolverá true si la sentencia SQL pasada como parámetro retorna algún resultado.</p> <p>Parámetros:</p> <p><i>\$SQL</i>: Sentencia SQL a ejecutar.</p>
Funciones para el conteo de registros	

<p>count_records(\$table, \$field1="", \$value1="", \$field2="", \$value2="", \$field3="", \$value3="")</p>	<p>Devolverá el número de registros que haya en la tabla \$table cuyos valores en los campos \$fieldx se correspondan con \$valuex.</p> <p>Parámetros:</p> <p>Análogos a los parámetros de la función record_exists().</p>
<p>count_records_select(\$table, \$select="", \$countitem='COUNT(*)')</p>	<p>Devolverá el número de registros que existen en la tabla \$table cuyos valores que se correspondan con el fragmento WHERE pasado como parámetro.</p> <p>Parámetros:</p> <p><i>\$table</i>: Tabla en la que buscar los registros.</p> <p><i>\$select</i>: Fragmento completo WHERE.</p> <p><i>\$countitem</i>: Elemento sobre el que realizar la cuenta.</p>
<p>count_records_sql(\$sql)</p>	<p>Devolverá el resultado de la ejecución de la sentencia SQL.</p> <p>Parámetros:</p> <p><i>\$SQL</i>: Sentencia SQL a ejecutar, del tipo 'SELECT COUNT(...) ...'.</p>
<p>Funciones para la obtención de registros</p>	
<p>Con un comportamiento similar a las funciones anteriores, en dmllib.php podemos encontrar, entre otras y junto con su documentación, las siguientes funciones para la obtención de registros:</p> <p>get_record() para obtener un único registro en forma de objeto.</p> <p>get_record_select() idem, pero introduciendo una cláusula where.</p> <p>get_record_sql() idem, pero introduciendo una sentencia SQL completa.</p> <p>get_records() para obtener un array de registros en forma de objetos.</p> <p>get_records_select() idem, pero introduciendo una cláusula where.</p> <p>get_records_sql() idem, pero introduciendo una sentencia SQL completa.</p>	

Funciones para el borrado de registros	
<p>Para el borrado de registros de la base de datos, únicamente se implementan dos funciones importantes:</p> <p>delete_records(), con parámetros análogos a record_exists(), borra todos los registros que tengan valores coincidentes en los campos pasados como parámetros.</p> <p>delete_records_sql() borra todos los registros de la tabla \$table pasada como parámetro que coinciden con la cláusula WHERE también pasada como parámetro.</p>	
Funciones para la inserción y actualización de registros	
<pre>insert_record(\$table, \$dataobject, \$returnid=true, \$primarykey='id')</pre>	<p>Inserta un registro pasado como objeto, cuyos nombres de atributos se corresponderán con los campos del registro, en la tabla pasada como parámetro.</p> <p>Parámetros:</p> <p><i>\$table</i>: Tabla en la que insertar el registro.</p> <p><i>\$dataobject</i>: Objeto conteniendo el registro a insertar.</p> <p><i>\$returnid</i>: Si es true, la función devolverá el identificador del nuevo registro insertado.</p> <p><i>\$primarykey</i>: La clave primaria del registro a insertar, en la mayoría de los casos es 'id'.</p>
<pre>update_record(\$table, \$dataobject)</pre>	<p>Actualiza un registro en la tabla cuyo nombre es pasado como parámetro.</p> <p>Parámetros:</p> <p><i>\$table</i>: Tabla en la que actualizar el registro.</p> <p><i>\$dataobject</i>: Objeto conteniendo el registro a insertar, cada nombre de atributo debe corresponderse con un campo de la tabla, y debe tener un campo 'id' que sirva para identificarlo en la tabla especificada.</p>
<p>Las funciones descritas previamente serán las que el desarrollador use en un 99% de los casos, para obtener documentación de funciones menos usuales, como de costumbre, observar la documentación del archivo.</p>	
weblib.php	Funciones para la creación de elementos

	html y salida web en general.
<p>A continuación se referencian algunas de las funciones que el desarrollador debería conocer antes de comenzar su trabajo y que, como siempre, se encuentran documentadas en el propio código fuente:</p> <p><code>print_heading()</code>, como sustituto del conocido <code>&lt;h1&gt;</code></p> <p><code>print_heading_with_help()</code>, idem, pero proporcionando un botón de ayuda.</p> <p><code>helpbutton()</code> imprime un botón de ayuda.</p> <p><code>print_simple_box()</code> imprime una caja en la que se puede colocar cualquier tipo de contenido.</p> <p><code>choose_from_menu()</code> imprime un menú de selección.</p> <p><code>choose_from_radio()</code> imprime botones de radio.</p> <p><code>print_textbox()</code> imprime una caja de texto.</p> <p><code>print_textfield()</code> imprime un campo de texto.</p> <p>...</p>	
<p>El contenido de este documento es muy extenso y resultará de gran ayuda al nuevo desarrollador, se aconseja leerlo detenidamente, prestando especial atención a la forma en la que Moodle facilita la creación de controles de formularios así como a las funciones que se encargan de recoger las variables de los mismos (<i><code>required_param()</code></i> y <i><code>data_submitted()</code></i> especialmente).</p> <p>Así como se especifica el uso obligatorio de la función <code>required_param()</code> (aunque el acceso a las variables <code>\$_GET</code> y <code>\$_POST</code> funciona perfectamente), en ningún documento de Moodle se obliga al uso de funciones para crear elementos del formulario, aunque facilita mucho la tarea al programador.</p>	
moodlelib.php	Funciones de propósito general para Moodle
<p>En este fichero pueden encontrarse todo tipo de funciones de propósito general para Moodle, conviene que el desarrollador novel las ojee antes de comenzar su trabajo sin llegar a prestarles una atención mayor de lo conveniente para no complicarse demasiado.</p> <p>Según vaya avanzando en el desarrollo, es posible que le surja la duda de si cierta funcionalidad, ligada fuertemente con el núcleo de Moodle, estará desarrollada. Seguramente lo esté aquí.</p>	



## 2.7 Ficheros de un módulo de actividad

Para poder realizar correctamente un módulo de actividades en Moodle, será necesario que cree alguno de los siguientes ficheros. Téngalos muy en cuenta a la hora de seguir un orden de creación.

En caso de necesitar consultar alguna de las funciones de Moodle acuda a la dirección: <http://xref.moodle.org/> aquí se encuentran todas ellas ordenadas de la misma manera que Moodle.

### 2.7.1 El fichero INSTALL.XML

Este fichero XML define las tablas de la base de datos que tendrá su módulo, establece una gramática para la creación de tablas y hacer así a Moodle más portable, frente las anteriores versiones, que definían un archivo para cada tipo de base de datos.

Install.XML define el tipo de datos, longitud, nombre y, en general, todos los campos que pueden definirse con sentencias SQL. La creación de las tablas se lleva a cabo con la instalación del módulo.

A continuación se adjunta y comenta un fichero de ejemplo:

```

<!-- Definición del tipo de documento, no se modifica -->
<?xml version="1.0" encoding="UTF-8" ?>
<XMLDB PATH="mod/nombremodulo/db" VERSION="20070401" COMMENT="Archivo XMLDB para
mod/nombremodulo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../../lib/xmldb/xmldb.xsd"
>
<!-- Definición de las tablas -->
<TABLES>
  <!-- Definición de la tabla del módulo, debe existir obligatoriamente
  En el caso de que haya más de una tabla, deben enlazarse mediante los campos
  PREVIOUS y NEXT, que apuntan a la tabla anterior y siguiente, respectivamente.
  -->
  <TABLE NAME=" nombremodulo " COMMENT="Comentario sobre la tabla nombremodulo ">
    <!-- Campos que tendrá la tabla -->
    <!--
      NAME: Nombre del campo
      TYPE: Tipo del dato
      LENGTH: Longitud del dato
      NOTNULL: El campo puede ser o no nulo
      UNSIGNED: El campo tiene signo o no
      SEQUENCE y ENUM: El campo es autoincremental o no
      COMMENT: Comentario sobre el campo
      NEXT: Siguiente campo de la tabla
      PREVIOUS: Campo previo de la tabla
    -->
    <FIELDS>
      <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
      SEQUENCE="true" ENUM="false" COMMENT="id de la instancia" NEXT="course"/>
      <FIELD NAME="course" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
      DEFAULT="0" SEQUENCE="false" ENUM="false" COMMENT="Curso al que pertenece la instancia"
      PREVIOUS="id" NEXT="name"/>
      <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="true" SEQUENCE="false"
      ENUM="false" COMMENT="Nombre de la instancia" PREVIOUS="course">
    <!-- El resto de campos de la tabla deben ir aquí -->
    </FIELDS>
    <!-- Claves de la tabla
      NAME: Nombre de la clave
      TYPE: Primary, unique o foreign
    -->
  </TABLE>
</TABLES>

```

```
    FIELDS: Campos que componen la clave
    COMMENT: Comentario sobre la clave
    OPCIONALES:
        NEXT y PREVIOUS: Apuntando a la clave siguiente y anterior, de existir.
        REFFIELDS Y REFTABLE: En el caso de las claves externas, campo y tabla al
que referencian
-->
<KEYS>
  <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Clave primaria para
nombremodulo " />
</KEYS>
<!-- Índices de la tabla, campos análogos a los de las claves -->
<INDEXES>
  <INDEX NAME="course" UNIQUE="false" FIELDS="course"/>
</INDEXES>
</TABLE>
</TABLES>
<!-- Sentencias SQL que se ejecutarán al instalar el módulo -->
<STATEMENTS>
  <!-- Sentencia
    Se define el nombre, el tipo de sentencia, la tabla sobre la que se ejecuta y
un comentario descriptivo -->
  <STATEMENT NAME="insert log_display" TYPE="insert" TABLE="log_display"
COMMENT="Inserta registros en la tabla de log" />
  <SENTENCES>
    <!-- Contenido de las sentencias propiamente dicho, no usan campos PREVIOUS ni
NEXT -->
    <SENTENCE TEXT="(module, action, mtable, field) VALUES (nombremodulo, 'add',
'nombremodulo', 'name')" />
    <SENTENCE TEXT="(module, action, mtable, field) VALUES (nombremodulo, 'update',
'nombremodulo', 'name')" />
  </SENTENCES>
</STATEMENT>
</STATEMENTS>
</XMLDB>
```

### 2.7.2 El fichero ACCESS.PHP

El fichero access.php define las capacidades de los usuarios en función a los roles para este módulo, su formato es el siguiente:

```
<?php
$mod_nombremodulo_capabilities = array(
    'mod/ nombremodulo:nombre_capacidad_primera' => array(
        ///Tipo de capacidad
        'captype' => 'write',
        ///Contexto de la actividad
        'contextlevel' => CONTEXT_MODULE,
        ///Permisos
        'legacy' => array(
            'editingteacher' => CAP_PROHIBIT,
            'coursecreator' => CAP_PROHIBIT,
            'teacher' => CAP_PROHIBIT,
            'admin' => CAP_PROHIBIT,
            'student' => CAP_ALLOW
        )
    ),
    ///Aquí se pueden añadir tantas capacidades como se desee
    'mod/nombremodulo:nombre_capaciad_enesima' => array(
        'captype' => 'write',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'editingteacher' => CAP_ALLOW,
            'coursecreator' => CAP_ALLOW,
            'teacher' => CAP_ALLOW,
            'admin' => CAP_ALLOW,
            'student' => CAP_PROHIBIT
        )
    )
);
?>
```

## 2.7.3 El fichero MOD.HTML

Se trata de un fichero html con código php embebido que es usado para la creación y configuración del módulo en el momento de la creación de una instancia del mismo.

Cada campo que defina la instancia de la actividad deberá ser un campo del objeto php **\$form**, de la forma *\$form->nombre del campo*. Será útil almacenar algunos de estos campos en las tablas de la base del dato correspondientes al módulo. El desarrollador no deberá preocuparse, de momento, por el procesamiento de estos datos, ya que los realizará el archivo mod.php, implementado previamente por Moodle y que el usuario no deberá modificar.

Para evitar fallos en el procesamiento del formulario, las variables deben ser inicializadas antes de su uso, compruebe esto al principio del fichero mediante trozos de código de php de la siguiente manera:

```
<?php
if (!isset($form->name)) {
    $form->name = '';
}
if (!isset($form->description)) {
    $form->description = '';
} //Resto de bloques similares
?>
```

Existen ciertos campos que deben aparecer obligatoriamente en cada actividad: **name**, **description**, **course** e **id** (aparecerán como atributos de \$form). Los dos primeros deben tomar valores introducidos por el usuario mediante un formulario web, los dos últimos los define automáticamente el sistema mediante el uso de campos de formulario del tipo hidden, junto con otros atributos que se enumeran a continuación, de los que se comentan los más importantes:

```
//El archivo que procesará el formulario
<form name="form" method="post" action="mod.php">
//Campos adicionales que el desarrollador desee establecer
//...
//Campos que establece automáticamente Moodle y que siempre se declaran de la misma
manera
//El campo course hace referencia al curso en el que se desarrolla la actividad
<input type="hidden" name="course" value="<?php p($form->course) ?>" />
//El campo sesskey hace referencia a la clave de la sesión actual
<input type="hidden" name="sesskey" value="<?php p($form->sesskey) ?>" />
<input type="hidden" name="coursemodule" value="<?php p($form->coursemodule) ?>" />
<input type="hidden" name="section" value="<?php p($form->section) ?>" />
<input type="hidden" name="module" value="<?php p($form->module) ?>" />
//Nombre del módulo (no de la instancia)
<input type="hidden" name="modulename" value="<?php p($form->modulename) ?>" />
//Número identificador de la instancia
<input type="hidden" name="instance" value="<?php p($form->instance) ?>" />
<input type="hidden" name="mode" value="<?php p($form->mode) ?>" />
</form>
```

El resto de campos que el desarrollador desee almacenar en la base de datos, como la fecha a partir de la cual la actividad estará disponible, o cualquier otro parámetro que desee que sea configurable, deberá ser definido en este fichero.

## 2.7.4 El fichero VIEW.PHP

El fichero view.php es, seguramente, junto con el fichero lib.php, uno de los más importantes a la hora de desarrollar un módulo de actividades para Moodle, en él se define el contenido que se mostrará a cada usuario de la actividad mediante el uso de capacidades (para más información, vea el apartado [El fichero ACCESS.PHP](#)).

A continuación se pasa a comentar el código mínimo que debe contener este fichero:

```
<?php
/// Reemplazar "nombremodulo" por el nombre del módulo a desarrollar
///Include que todos los ficheros php deben tener
require_once("../../config.php");
require_once("lib.php");
///Estas líneas se establecen por defecto para obtener el id de la instancia actual
$id = optional_param('id', 0, PARAM_INT);
$a = optional_param('a', 0, PARAM_INT);
///Haciendo comprobaciones sobre los parámetros pasados
if ($id) {
    if (!$cm = get_record("course_modules", "id", $id)) {
        error("Course Module ID was incorrect");
    }
    if (!$course = get_record("course", "id", $cm->course)) {
        error("Course is misconfigured");
    }
    if (!$nombremodulo = get_record("nombremodulo", "id", $cm->instance)) {
        error("Course module is incorrect");
    }
} else {
    if (!$nombremodulo = get_record("nombremodulo", "id", $a)) {
        error("Course module is incorrect");
    }
    if (!$course = get_record("course", "id", $nombremodulo ->course)) {
        error("Course is misconfigured");
    }
    if (!$cm = get_coursemodule_from_instance("nombremodulo ", $nombremodulo ->id,
$course->id)) {
        error("Course Module ID was incorrect");
    }
}
///Para ver esta página, el usuario debe estar autenticado
require_login($course->id);
///Añade al log la visita de la actividad
add_to_log($course->id, " nombremodulo ", "view", "view.php?id=$cm->id", "$
nombremodulo ->id");
/// Para imprimir el encabezado de la página
if ($course->category) {
    $navigation = "<a href=\"../../course/view.php?id=$course->id\">$course-
>shortname</a> ->";
} else {
    $navigation = '';
}
///Ver documentación de la función get_string()
$strnombremodulos = get_string("modulenameplural", " nombremodulo ");
$strnombremodulo = get_string("modulename", " nombremodulo ");
///Impresión del encabezado propiamente dicho
print_header("$course->shortname: $ nombremodulo ->name", "$course->fullname",
"$navigation <a href=index.php?id=$course->id>$strnombremodulos</a> ->
$ nombremodulo ->name",
"", "", true, update_module_button($cm->id, $course->id,
$strnombremodulos),
navmenu($course, $cm));
/// Parte principal de la página
///Aquí debe ir el código del desarrollador
/// Imprime el pié de página
print_footer($course);
?>
```

## 2.7.5 El fichero LIB.PHP

Lib.php es la biblioteca de funciones del módulo a desarrollar, en ella se implementará toda (o casi toda) la funcionalidad del módulo. Existen un conjunto de normas que se deben seguir a la hora de desarrollar esta biblioteca de las que destacamos las siguientes:

- Las funciones deben nombrarse de la siguiente manera: `nombremodulo_nombre_funcion()`.
- Las variables globales deberán nombrarse de la siguiente manera: `$NOMBREMODULO_NOMBRE_VARIABLE`.

Lib.php deberá implementar ciertas funciones preestablecidas que usará Moodle con distintos propósitos:

- `nombremodulo_add_instance($nombremodulo)`
- `nombremodulo_update_instance($nombremodulo)`
- `nombremodulo_delete_instance($id)`
- `nombremodulo_user_outline($course, $user, $mod, $nombremodulo)`
- `nombremodulo_user_complete($course, $user, $mod, $nombremodulo)`
- `nombremodulo_print_recent_activity($course, $isteacher, $timestart)`
- `nombremodulo_cron()`
- `nombremodulo_grades($nombremoduloid)`
- `nombremodulo_get_participants($nombremoduloid)`
- `nombremodulo_scale_used ($nombremoduloid,$scaleid)`

A continuación se detallará qué líneas debe seguir la implementación de cada una de estas funciones:

```
<?php
$NOMBREMODULO_CONSTANT = 7;      ///Ejemplo de variable global
/**
 * Dado un objeto que contenga toda la información necesaria,
 * (definida por el formulario de mod.html) esta función
 * creará una nueva instancia y retornará el identificador de la misma
 * Todas las modificaciones que se quieran hacer a los datos del formulario
 * antes de introducirlos en la base de datos (por ejemplo, comprobación de valores)
 * deberán hacer aquí
 * @param object $nombremodulo Un objeto definido en el formulario de mod.html
 * @return int El identificador de la instancia
 */
function nombremodulo_add_instance($nombremodulo) {
    ///Implementar la funcionalidad que sea necesaria
    return insert_record("nombremodulo", $nombremodulo);
}
/**
 * Dado un objeto que contenga toda la información necesaria,
 * (definida por el formulario de mod.html) esta función
```

```

* actualizará la instancia cuya ID de corresponde con el atributo id de $nombremodulo
* Todas las modificaciones que se quieran hacer a los datos del formulario
* antes de introducirlos en la base de datos (por ejemplo, comprobación de valores)
* deberán hacer aquí.
* Funcionamiento análogo a nombremodulo_add_instance()
* @param object $nombremodulo Un objeto definido en el formulario de mod.html
* @return boolean True si la operación se lleva a cabo exitosamente
**/
function nombremodulo_update_instance($nombremodulo) {
    ///Es necesario que el objeto a actualizar tenga un campo id
    $nombremodulo->id = $nombremodulo->instance;
    return update_record("nombremodulo", $nombremodulo);
}
/**
* Dado un identificador de una instancia de este módulo,
* esta función borrará todos los registros asociados con la misma,
* así como todos los datos asociados a ella.
* @param int $id Identificador de la instancia
* @return boolean True si la operación se lleva a cabo exitosamente
**/
function nombremodulo_delete_instance($id) {
    ///Obtenemos la instancia
    if (! $nombremodulo = get_record("nombremodulo", "id", "$id")) {
        return false;
    }
    $result = true;
    ///Borramos todos los registros necesarios, como mínimo el de la instancia
    if (! delete_records("nombremodulo", "id", "$nombremodulo->id")) {
        $result = false;
    }
    return $result;
}
/**
* Dado un curso y un usuario, devuelve un objeto con información somera
* respecto a las actividades que el usuario ha llevado a cabo con esta
* instancia.
* Esta función es usada para reportes de la actividad, su implementación
* no es necesaria si no se desea mostrar reportes de actividad.
* Ejemplo de retorno:
* $return->time = El momento en el que sucedió el evento
* $return->info = Un pequeño texto descriptivo
* @param int $course Identificador del curso al que pertenece la instancia
* @param int $user Identificador del usuario cuya información deseamos obtener
* @param object $mod Objeto que contiene información sobre el módulo
* @param object $nombremodulo Objeto que contiene información sobre la instancia
* @return object Descrito más arriba.
**/
function nombremodulo_user_outline($course, $user, $mod, $nombremodulo) {
    return $return;
}
/**
* Dado un curso y un usuario, imprime información detallada
* respecto a las actividades que el usuario ha llevado a cabo con esta instancia.
* Esta función es usada para reportes de la actividad, su implementación
* no es necesaria si no se desea mostrar reportes de actividad.
* @param int $course Identificador del curso al que pertenece la instancia
* @param int $user Identificador del usuario cuya información deseamos obtener
* @param object $mod Objeto que contiene información sobre el módulo
* @param object $nombremodulo Objeto que contiene información sobre la instancia
* @return boolean True si ha habido alguna salida
**/
function nombremodulo_user_complete($course, $user, $mod, $nombremodulo) {
    return true;
}
/**
* Dado un curso y un tiempo, esta función buscará información
* cercana temporalmente a ese instante y la imprimirá
* Esta función puede no implementarse si no se considera oportuno
* @uses $CFG
* @param int $course Identificador del curso al que pertenece la instancia
* @param boolean $isteacher True si la persona que desea ver la información es profesor
* @param int $timestart Instante de tiempo de referencia
* @return boolean True si ha habido alguna salida
**/
function nombremodulo_print_recent_activity($course, $isteacher, $timestart) {
    global $CFG;

```

```

    return false;
}
/**
 * Esta función se ejecutará automáticamente cada vez que corra el servicio cron
 * Todas las actividades que se realicen periódicamente (actualización de datos,
 * envío de correos, etc.), deberán ponerse aquí.
 * Esta función puede no implementarse si no se considera oportuno
 * @uses $CFG
 * @return boolean True si la operación ha sido exitosa
 */
function nombremodulo_cron() {
    global $CFG;
    return true;
}
/**
 * Para actividades evaluables, debe devolver un objeto que contenga las
 * notas para cada estudiante, así como la nota máxima obtenible.
 * Ejemplo de retorno:
 *   $return->grades = array de notas, indexadas por el ID del usuario;
 *   $return->maxgrade = Nota máxima posible;
 *   return $return;
 * @param int $nombremoduloid Identificación de la instancia del módulo
 * @return mixed Null o un objeto conteniendo las notas y la nota máxima
 */
function nombremodulo_grades($nombremoduloid) {
    return NULL;
}
/**
 * Devolverá un array de registros con toda la información de los usuarios participantes
 * de la instancia. Incluirá a todos los usuarios implicados, también profesores
 * @param int $nombremoduloid Identificación de la instancia del módulo
 * @return mixed boolean o array de estudiantes
 */
function nombremodulo_get_participants($nombremoduloid) {
    return false;
}
/**
 * Esta función retornará si una escala (de evaluación) está siendo usada
 * por el módulo, en caso de que tenga soporte para notas y escalas.
 * Esta función puede no implementarse si no se considera oportuno
 * Observar la documentación de los módulos forum, glossary y journal como referencia
 * @param int $nombremoduloid Identificación de la instancia del módulo
 * @param int $scaleid Identificación de la escala
 * @return mixed Ver documentación de forum, glossary o journal
 */
function nombremodulo_scale_used ($nombremoduloid,$scaleid) {
    $return = false;
    //$rec = get_record("nombremodulo","id",$nombremoduloid,"scale","-$scaleid");
    //
    //if (!empty($rec) && !empty($scaleid)) {
    //    $return = true;
    //}
    return $return;
}
/// Aquí irán el resto de funciones que sea necesario implementar en el módulo.
?>

```

## 2.7.6 El fichero INDEX.PHP

El fichero index.php listará todas las instancias en el curso actual del módulo que esté desarrollando. Aquí se podrá mostrar la información que se crea oportuna de cada instancia.

A continuación se comenta un archivo de ejemplo, las líneas no comentadas son líneas estándar que no es necesario modificar:

```

<?php
    require_once("../../config.php");
    require_once("lib.php");
    $id = required_param('id', PARAM_INT); // Obtención del curso
    if (! $course = get_record("course", "id", $id)) {

```

```

        error("Course ID is incorrect");
    }
    require_login($course->id);
    //Añade al log del curso
    add_to_log($course->id, "nombremodulo", "view all", "index.php?id=$course->id", "");
    $strnombremodulos = get_string("modulenameplural", "nombremodulo");
    $strnombremodulo = get_string("modulename", "nombremodulo");
    //Impresión de la cabecera
    if ($course->category) {
        $navigation = "<a href=\"../../course/view.php?id=$course->id\">$course-
>shortname</a> ->";
    } else {
        $navigation = '';
    }
    print_header("$course->shortname: $strnombremodulos", "$course->fullname",
"$navigation $strnombremodulos", "", "", true, "", navmenu($course));
    //Obtiene toda la información de las instancias
    if (!$nombremodulos = get_all_instances_in_course("nombremodulo", $course)) {
        notice("There are no nombremodulos", "../../course/view.php?id=$course->id");
        die;
    }
    // Imprime toda la información de cada instancia, que puede ampliarse como se crea
necesario
    $timenow = time();
    $strname = get_string("name");
    $strweek = get_string("week");
    $strtopic = get_string("topic");
    if ($course->format == "weeks") {
        $table->head = array ($strweek, $strname);
        $table->align = array ("center", "left");
    } else if ($course->format == "topics") {
        $table->head = array ($strtopic, $strname);
        $table->align = array ("center", "left", "left", "left");
    } else {
        $table->head = array ($strname);
        $table->align = array ("left", "left", "left");
    }
    foreach ($nombremodulos as $nombremodulo) {
        if (!$nombremodulo->visible) {
            //Si la instancia no es visible, lo muestras más oscuro
            $link = "<a class=\"dimmed\" href=\"view.php?id=$nombremodulo-
>coursemodule\">$nombremodulo->name</a>";
        } else {
            //Si es visible, lo muestra normal
            $link = "<a href=\"view.php?id=$nombremodulo->coursemodule\">$nombremodulo-
>name</a>";
        }
        //Solo mostrará la información si el formato del curso es semanal o por temas
        if ($course->format == "weeks" or $course->format == "topics") {
            $table->data[] = array ($nombremodulo->section, $link);
        } else {
            $table->data[] = array ($link);
        }
    }
    echo "<br />";
    //Impresión de la información
    print_table($table);
    //Impresión del pié de página
    print_footer($course);
?>

```

## 2.7.7 El fichero VERSION.PHP

Se trata de un fichero que únicamente incluye el número de versión del módulo que está tratando, usado para controlar las actualizaciones entre nuevas versiones, su formato es el siguiente:

```

<?php
$module->version = 2007040200; // Versión actual del módulo (Formato YYYYMMDDX)
$module->cron = 0; // Intervalo de tiempo, en segundos, en el que cron
deberá

```

```
?> // comprobar la versión de este módulo
```

## 2.7.8 El fichero UPGRADE.PHP

Este fichero mantiene un registro de los cambios que es necesario realizar entre distintas versiones del módulo.

En ocasiones la actualización de una antigua versión a una nueva puede implicar cambios mayores (como por ejemplo, en la base de datos) que pueden hacer que el módulo deje de funcionar correctamente.

Los comandos que se usen en este fichero deben ser neutrales en cuanto a la base de datos, usando las funciones definidas en lib/ddlib.php.

La función “upgrade” deberá devolver verdadero si todos los cambios se han llevado a cabo de manera satisfactoria.

```
<?php
function xmldb_nombremodulo_upgrade($oldversion=0) {
    global $CFG, $THEME, $db;
    $result = true;
    // Primer ejemplo, actualización necesaria si la versión actual es
    // anterior a la del 01 de Abril de 2008
    if ($result && $oldversion < 2008040100) {
        //Introducir actualizaciones a realizar
    }
    // Primer ejemplo, actualización necesaria si la versión actual es
    // anterior a la del 12 de Abril de 2008
    if ($result && $oldversion < 2008041200) {
        //Introducir actualizaciones a realizar
    }
    return $result;
}
?>
```

## 2.7.9 Los ficheros de idioma

Moodle proporciona un sistema de internacionalización bastante útil y sencillo que consiste en un fichero por cada idioma, localizado dentro de la carpeta /lang/nombre\_idioma/ y que recibe el nombre del módulo seguido de la extensión .php (en este caso, nombremodulo.php)

La estructura del fichero es la siguiente:

```
<?php
//El índice del array será la cadena que desea sustituir,
// El contenido será la cadena por la que la sustituirá.
//Las tres siguientes cadenas deben establecerse siempre
$string['nombremodulo'] = 'nombremodulo';
$string['modulename'] = 'Nombre del módulo';
$string['modulenameplural'] = 'Nombre de los módulos';
//A continuación, todas las cadenas que use su módulo
$string['nombremoduloejemplo'] = 'Ejemplo de cadena';
?>
```

Para acceder al contenido de los ficheros de idioma simplemente tendrá que usar las funciones `get_string` o `print_string`, la primera de ellas devuelve el contenido de una cadena, la segunda lo imprime:

```
///Con la siguiente línea obtendrá el valor de nombrecadena
// en el idioma que el usuario haya seleccionado
get_string('nombrecadena', 'nombremodulo');
///Con la siguiente línea imprimirá el valor de nombrecadena
// en el idioma que el usuario haya seleccionado
print_string('nombrecadena', 'nombremodulo');
```

Moodle proporciona la posibilidad de pasar algún parámetro a los ficheros de idiomas de la siguiente manera:

```
///Paso de un único parámetro
/// Tipo de entrada en el fichero de idioma:
$string['nombremoduloejemplo'] = 'Ejemplo de cadena con $a';
///Obtención de la cadena:
$a= 'parámetro';
///La siguiente línea devolverá "Ejemplo de cadena con parámetro":
get_string('nombremoduloejemplo', 'nombremodulo', $a);
///La siguiente línea imprimirá "Ejemplo de cadena con parámetro":
print_string('nombremoduloejemplo', 'nombremodulo', $a);
/// Paso de varios parámetros
/// Tipo de entrada en el fichero de idioma:
$string['nombremoduloejemplo'] = 'Ejemplo de cadena con $a[0] y $a[1]';
///Obtención de la cadena:
$a[0]= 'parámetro 0';
$a[1]= 'parámetro 1';
///La siguiente línea devolverá "Ejemplo de cadena con parámetro 0 y parámetro 1":
get_string('nombremoduloejemplo', 'nombremodulo', $a);
///La siguiente línea imprimirá "Ejemplo de cadena con parámetro 0 y parámetro 1":
print_string('nombremoduloejemplo', 'nombremodulo', $a);
```

### 2.7.10 Los ficheros de Ayuda

Moodle permite la creación de ficheros de ayuda para el usuario en formato html mediante un sistema similar al de los ficheros de idioma que permite una creación y manipulación sencilla de cada elemento de ayuda.

Los ficheros de ayuda son ficheros html que se incluyen dentro un directorio con el nombre del módulo incluido a su vez en el directorio help del directorio de lenguaje (en definitiva: lang/idioma/help/nombremodulo/)

Cada tema de ayuda debe incluirse en un único fichero html que deberá ser enlazado desde el fichero index.html de la siguiente manera:

```
<p>Nombre del módulo</p>
<ul>
  <li><a href="help.php?module=newmodule&amp;file=mods.html">Información
general</a></li>
  <li><a href="help.php?module=newmodule&amp;file=tema1.html">Tema 1</a></li>
  <li><a href="help.php?module=newmodule&amp;file=tema2.html">Tema 2</a></li>
</ul>
```

El fichero mods.html debe contener información general sobre el módulo.

Estos temas de ayuda serán listados directamente en el archivo de temas de ayuda de Moodle, no obstante, el desarrollador puede enlazarlos en lugar en el que crea oportuno mediante funciones como helpbutton() o print\_heading\_with\_help()

```
helpbutton('fichero_sin_extension', 'Título de la ventana', 'nombremodulo');
print_heading_with_help('Encabezado', 'fichero_sin_extension', 'nombremodulo');
```





### 3 Referencias Bibliográficas

- [González07] González Benito, Guillermo; “Proyecto: Desarrollo e integración en la plataforma Moodle de módulos de apoyo para la adaptación de asignaturas al Espacio Europeo de Educación Superior”. Universidad de Oviedo. 2007
- “Página Principal – MoodleDocs”. <http://docs.moodle.org/es> . 2009
- “Cómo instalar Moodle 1.8.1 en Ubuntu”. <http://www.ubuntu-es.org/?q=node/53465> 2007
- [Meroño07]. Meroño Hernández, Alejandro. “Instalar XAMPP en Ubuntu”. <http://www.alejandrox.com/2007/03/instalar-xampp-en-ubuntu/>. 2007
- [Meroño07]. Meroño Hernández, Alejandro. “Instalar Moodle en Ubuntu”. <http://www.alejandrox.com/2007/03/instalar-moodle-en-ubuntu/>. 2007
- “Hablemos de Moodle”. <http://blog.pucp.edu.pe/item/3747>. 2006
- [BurónArtillesGarcíaFidalgoRubioCastro06]. Burón Fernández, Francisco Javier; Artilles Larralde Alfredo; García Salcines, Enrique; Fidalgo Blanco, F. Angel; Rubio Royo, Enrique; Castro Lozano, Carlos. “e-aprendo, Virtual Learning Management based on Moodle”. <http://www.formatex.org/micte2006/pdf/40-44.pdf>. 2006
- [ClerenciaPérez]. Pérez Clerencia, Isaac; Pérez Oñate, Borja. “Introducción al Moodle. Universidad de Zaragoza”. <http://moodle.unizar.es/>.
- [ClerenciaPérez]. Pérez Clerencia, Isaac; Pérez Oñate, Borja. “Moodle- manual del profesor”. <http://moodle.unizar.es/file.php/1/Manual-profesor-moodle.pdf>
- [Rodríguez07]. Rodríguez Martín, Francisco Javier. “Tutorial para la creación de un módulo en Moodle”. [http://www.moodle.org/file.php/11/moddata/forum/338/366774/Tutorial\\_M\\_dulos.pdf](http://www.moodle.org/file.php/11/moddata/forum/338/366774/Tutorial_M_dulos.pdf). 2007
- [Castro04]. Castro López- Tarruella, Enrique. “Moodle: Manual de usuario”. [http://moodle.org/file.php/11/manual\\_del\\_estudiante/Manual-usuario.pdf](http://moodle.org/file.php/11/manual_del_estudiante/Manual-usuario.pdf) . 2004
- [Martín06]. Martín Gómez, Jesús. “Moodle 1.5- Manual de consulta”. [http://moodle.org/file.php/11/manual\\_del\\_estudiante/Manual-usuario.pdf](http://moodle.org/file.php/11/manual_del_estudiante/Manual-usuario.pdf) . 2006
- EFMD-STEI-i. “Manual de Moodle para el alumno”. <http://www.steiformacio.com/imgs/ManualEFMDcastellanoR.pdf>
- Universidad Politécnica de Cartagena. “Manual del alumno- Aula virtual Moodle” <http://moodle.upct.es/file.php/1/manualalumno.pdf> .2009

- [Castillo07]. Castillo, Williams. “Manual de usuario del Campus Virtual de la DEP- FE mediante la plataforma Moodle”. <http://depfe-edu.economia.unam.mx/cursos/mod/resource/view.php?id=470> . 2007
- GATE. “Manual Moodle”. <http://cvsp.cucs.udg.mx/drupal6/documentos/manualmoodle.pdf>
- [Bournissen07]. Bournissen, Juan Manuel. “Manual del alumno. Una introducción al Campus virtual de Moodle”. <http://moodle.instituto20.com.ar/mod/resource/view.php?id=199> . 2007
- [ÁlvarezGonzálezLorenzoTabasco05]. Álvarez García, Rafael; González Conejero, Juan Enrique; Lorenzo Díaz, Francisco Manuel; Tabasco Guzmán, Carlos. “Moodle. Manual de usuario”. [http://virtual.uca.es/portalFormacion/docs/desarrolloMoodle/carpetaLocal3/plfng\\_view](http://virtual.uca.es/portalFormacion/docs/desarrolloMoodle/carpetaLocal3/plfng_view) . 2005
- Bakkali, Imane. “Guía de Moodle Mundusfor para administradores”. [http://cevug.ugr.es/mundusfor/file.php/1/Guia\\_de\\_Moodle\\_Mundusfor\\_para\\_administradores.pdf](http://cevug.ugr.es/mundusfor/file.php/1/Guia_de_Moodle_Mundusfor_para_administradores.pdf)