

Proyecto Fin de Carrera
Ingeniería en Informática

Sincronización del Sistema de Información de la ULPGC con Moodle

Autor:

Víctor Déniz Falcón

Tutor:

Alexis Quesada Arencibia

Escuela de Ingeniería Informática

Universidad de Las Palmas de Gran Canaria

Febrero de 2015

A Jenni, llenas mi vida y me haces desear que llegue la noche para recogerme en tus brazos, sin importar como haya ido el día; no dejo de aprender contigo y quiero seguir haciéndolo el resto de mi vida.

A Bianca y Aarón, me habéis enseñado otra forma de amar, otra forma de sentir. Habéis dado un revolcón a mi vida y ya no puedo imaginarla sin vosotros.

A Mawi, tu sacrificio y esfuerzo me permitió realizar unos estudios universitarios, aun cuando no respondía como te merecías. Has dado parte de tu vida para que yo tenga la mía. Nunca podré agradecerte lo suficiente su entrega.

A Papá, Mamé y Lali, sin ustedes no sería quién soy. No me hace falta preguntar, sé que siempre están ahí.

Los quiero...

Agradecimientos

Durante los años que me ha llevado realizar el proyecto final de carrera, por diferentes motivos de índole personal y profesional he empleado muchos más de los necesarios, son demasiadas las personas a las que tengo que agradecer en mayor o menor medida su aportación.

A todos los tutores que en algún momento he perturbado, más bien poco, con mi intención de hacer el proyecto. En especial a Alexis Quesada, cuya cercanía y atención, a pesar de mi irregularidad, me ha marcado el camino a seguir.

A mis compañeros del Servicio de Informática de la ULPGC, especialmente a Luci, que entre café y queque siempre han puesto a mi disposición los recursos necesarios y han facilitado mi trabajo.

A todos mis profesores, desde la guardería hasta el último curso de carrera, por el conocimiento transmitido y vuestra aportación en mi persona. No puedo dejar de mencionar a Don Manuel, su convicción y metodología me marcó profundamente.

Resumen

Se ha desarrollado un conjunto de extensiones para el LMS (*Learning Management System*) Moodle que permiten sincronizar su contenido con la información académica de la ULPGC, almacenada en una base de datos institucional. De esta forma se minimiza la intervención humana, garantizando que el Campus Virtual de la ULPGC sea un reflejo de su estructura presencial, incluyendo titulaciones, asignaturas, profesores y alumnos.

Para ello se han seguido las directrices marcadas por los desarrolladores de Moodle, respetando la arquitectura de este software y utilizando la API que incorpora. Asimismo, las extensiones se han desarrollado con la vista puesta en su uso por terceros, por lo que, con pequeños cambios de configuración, se pueden utilizar en cualquier instalación de Moodle con características similares a las de la ULPGC.

Palabras Clave

Moodle, Desarrollo de extensiones de Moodle, Sincronización de sistemas, Campus Virtual de la ULPGC, *Learning Management System*, Extensión de identificación, Extensión de matriculación, Sincronización de bases de datos

Índice

Agradecimientos	iii
Resumen	v
Palabras Clave	vii
Índice.....	ix
Índice de Figuras	xv
Índice de Tablas.....	xvii
1 Introducción	1
1.1 <i>Justificación del Proyecto</i>	1
1.2 <i>Estado actual</i>	2
1.3 <i>Objetivos del Proyecto</i>	2
1.4 <i>Metodología</i>	3
1.4.1 <i>Metodología en Cascada</i>	3
1.5 <i>Planificación</i>	5
1.6 <i>Recursos necesarios</i>	8
1.7 <i>Estructura de la memoria</i>	8
2 E-learning	11
2.1 <i>E-learning on-line</i>	11
2.2 <i>Sistemas Gestión de Aprendizaje (LMS)</i>	12
2.2.1 <i>Usuarios</i>	13
2.2.2 <i>Características</i>	13
2.2.3 <i>Funciones</i>	14
2.3 <i>E-learning en la ULPGC</i>	15
2.4 <i>Organización del Campus Virtual de la ULPGC</i>	18

3 Moodle.....	19
3.1 <i>Organización interna de Moodle.....</i>	19
4 Sistema de información de la ULPGC	24
4.1 <i>Orígenes</i>	24
4.1.1 Servicio de Gestión Académica	25
4.1.2 Vicerrectorado de Ordenación Académica y EEES	25
4.2 <i>Servicio de Identificación Centralizada</i>	26
4.3 <i>Base de datos institucional: Oracle</i>	26
4.4 <i>Arquitectura física del Campus Virtual de la ULPGC.....</i>	27
4.4.1 Conexión Internet e Intranet.....	27
4.4.2 Servidores	28
4.4.3 Arquitectura de sistemas del Campus Virtual.....	30
5 Herramientas y tecnologías.....	31
5.1 <i>Subversion</i>	31
5.1.1 Repositorio.....	31
5.2 <i>Eclipse.....</i>	31
5.3 <i>Entorno de desarrollo.....</i>	32
5.3.1 PHP	32
5.3.2 MySQL y SQL.....	32
6 Desarrollo en Moodle	34
6.1 <i>Arquitectura de Moodle</i>	34
6.1.1 Entorno de ejecución	34
6.1.2 Capas de Moodle.....	35
6.1.3 Moodle como sistema modular	37
6.1.4 Núcleo de Moodle.....	39
6.2 <i>Organización del código.....</i>	40
6.3 <i>Base de datos.....</i>	41
6.3.1 La clase <i>moodle_database</i>	41
6.4 <i>Estilo del código.....</i>	42
6.5 <i>Seguridad</i>	43

6.6	<i>Internacionalización</i>	43
6.7	<i>Rendimiento</i>	43
7	Análisis	45
7.1	<i>Definición del Sistema</i>	45
7.1.1	Determinación del Alcance del Sistema	45
7.2	<i>Requisitos del Sistema</i>	46
7.2.1	Centros.....	46
7.2.2	Titulaciones.....	46
7.2.3	Asignaturas	47
7.2.4	Grupos.....	48
7.2.5	Usuarios	50
7.2.6	Matrículas	51
7.2.7	Global	52
7.3	<i>Identificación y descripción de los Subsistemas</i>	53
7.3.1	Sistema de Información de la ULPGC	53
7.3.2	Moodle.....	55
7.3.3	Descripción de los Interfaces entre Subsistemas	67
7.4	<i>Especificación del Plan de Pruebas</i>	69
7.4.1	Casos de prueba	70
8	Diseño del Sistema	77
8.1	<i>Arquitectura de sistemas</i>	77
8.1.1	Comunicación con el Sistema de información académico.....	77
8.1.2	Comunicación con el Sistema de identificación (CAS).....	78
8.2	<i>Componentes del sistema de sincronización</i>	78
8.2.1	Elementos comunes a todas las extensiones.....	79
8.2.2	Extensión local: sincronización de centros, titulaciones, asignaturas, grupos, asignación de grupos.....	81
8.2.3	Extensión de identificación	89
8.2.4	Extensión de matriculación	94
8.2.5	Secuencia de ejecución.....	96
8.3	<i>Especificación Técnica del Plan de Pruebas</i>	97

8.3.1	Pruebas Unitarias	97
8.3.2	Pruebas de Integración y del Sistema.....	100
9	Implementación del Sistema.....	101
9.1	<i>Preparación del entorno</i>	<i>101</i>
9.2	<i>Conexión a base de datos institucional.....</i>	<i>102</i>
9.3	<i>Auditoría.....</i>	<i>103</i>
9.4	<i>Extensión local</i>	<i>104</i>
9.4.1	Estructura de archivos	104
9.4.2	Estructura de los scripts.....	106
9.5	<i>Extensión de identificación</i>	<i>109</i>
9.5.1	Configuración de la extensión	110
9.5.2	Autenticación.....	111
9.5.3	Sincronización con la base de datos externa	113
9.6	<i>Extensión de matriculación.....</i>	<i>114</i>
9.6.1	Sincronización con la base de datos externa	115
9.7	<i>Ejecución completa de la sincronización.....</i>	<i>116</i>
10	Ejecución de las pruebas	117
10.1	<i>Realización de las pruebas</i>	<i>118</i>
10.1.1	Centros	118
10.1.2	Titulaciones	119
10.1.3	Asignaturas.....	120
10.1.4	Grupos	121
10.1.5	Usuarios.....	123
11	Conclusiones y líneas de trabajo futuras.....	125
11.1	<i>Conclusiones.....</i>	<i>125</i>
11.2	<i>Líneas de trabajo futuras.....</i>	<i>126</i>
12	Bibliografía.....	129
Anexo I: Instalación de Moodle		130
	<i>Código de Moodle</i>	<i>130</i>
	<i>Crear una base de datos</i>	<i>130</i>

<i>Crear directorio de datos</i>	131
<i>Instalación</i>	131
Anexo II: Subversion	132
<i>Creación del directorio de trabajo</i>	132
<i>Flujo de trabajo general</i>	132
Anexo III: Instalación y configuración de Eclipse	134
<i>Instalar Máquina virtual de Java (JVM, Java virtual machine)</i>	134
<i>Instalar Eclipse</i>	134
Anexo IV: Guía de estilo de código	137
<i>Herramientas</i>	137
<i>Formato de archivo</i>	137
<i>Convenciones de nombres</i>	139
<i>Arrays</i>	143
<i>Clases</i>	144
<i>Funciones y métodos</i>	145
<i>Estructuras de control</i>	146
<i>Documentación y comentarios</i>	148
<i>Excepciones</i>	157
<i>Funciones y constructores peligrosos</i>	158
<i>Código SQL</i>	158
Anexo V: Guía de seguridad	160
<i>Seguridad en las aplicaciones web</i>	160
<i>Diseño de seguridad de Moodle</i>	161
<i>Resumen de las directrices de seguridad</i>	163
Anexo VI Base de datos: guía de estilo y XMLDB	164
<i>Guía de estilo</i>	164
<i>Editor XMLDB</i>	166
Anexo VII: Manuales del sistema	168
<i>Manual de Instalación</i>	168
<i>Manual de Usuario</i>	169

Índice de Figuras

Figura 1-1 Fases de la metodología en Cascada.....	4
Figura 1-2 Planificación	8
Figura 2-1 Principales hitos e-learning en la ULPGC	17
Figura 3-1 Estructura de Moodle.....	20
Figura 3-2 Jerarquía de categorías y cursos.....	21
Figura 3-3 Diferencias entre identificación y matriculación.....	22
Figura 4-1 Esquema ULPnet.....	28
Figura 4-2 Infraestructura de las plataformas del Campus Virtual	29
Figura 4-3 Arquitectura de sistemas de la ULPGC.....	30
Figura 6-1 Entorno de ejecución	35
Figura 8-1 Sistemas involucrados	77
Figura 8-2 Tabla de centros.....	82
Figura 8-3 Tablas de Moodle implicadas en la sincronización	83
Figura 8-4 Diagrama de flujo de datos general scripts extensión local	84
Figura 8-5 Diagrama de flujo de datos Insertar Centro	85
Figura 8-6 Diagrama de flujo de datos Eliminar Centro.....	85
Figura 8-7 Diagrama de flujo de datos Insertar Categoría.....	86
Figura 8-8 Diagrama de flujo de datos Eliminar Categoría.....	86
Figura 8-9 Diagrama de flujo de datos Insertar Curso	86
Figura 8-10 Diagrama de flujo de datos Eliminar Curso	87
Figura 8-11 Diagrama de flujo de datos Insertar Grupo	87
Figura 8-12 Diagrama de flujo de datos Eliminar Grupo	87
Figura 8-13 Diagrama de flujo de datos Asignación Grupos	88
Figura 8-14 Diagrama de flujo de datos Asignar Grupo.....	89
Figura 8-15 Diagrama de flujo de datos Desasignar Grupo.....	89

Figura 8-16 Diagrama de secuencia del proceso de identificación	90
Figura 8-17 Diagrama de clases extensión Identificación	91
Figura 8-18 Tabla de usuarios de Moodle.....	94
Figura 8-19 Diagrama de clases extensión de matriculación	95
Figura 8-20 Tablas relacionadas con la matriculación	95
Figura 8-21 Secuencia de ejecución de los scripts de sincronización	96
Figura 9-1 Entorno de desarrollo.....	102
Figura 9-2 Conexión a base de datos externa.....	103
Figura 9-3 Estructura de archivos de extensión local	105
Figura 9-4 Estructura de archivos extensión identificación.....	110
Figura 9-5 Estructura de archivos extensión matriculación	115
Figura VII-1 Activación de extensión de identificación	170
Figura VII-2 Configuración de la extensión de identificación	171
Figura VII-3 Activación de la extensión de matriculación	172

Índice de Tablas

Tabla 1 Planificación	7
Tabla 2 Plataformas del Campus Virtual de la ULPGC	18
Tabla 3 Organización del código fuente.....	36
Tabla 4 Organización del directorio de datos.....	37
Tabla 5 Ejemplos de extensiones	39

1 INTRODUCCIÓN

1.1 JUSTIFICACIÓN DEL PROYECTO

La enseñanza a distancia (*e-learning*) dota de libertad para escoger cuándo y dónde estudiar, adaptándose al estilo de vida del alumno, permitiéndole compatibilizar su trabajo y tareas cotidianas con el aprendizaje de estudios universitarios.

Moodle es una plataforma de *e-learning* de distribución libre, con más de 45000 organizaciones y 20 millones de usuarios que hacen de este sistema uno de los más usados y con mayor crecimiento.

La Universidad de Las Palmas de Gran Canaria (ULPGC), desde el curso académico 2005/06, ofrece titulaciones de grado en modalidad no presencial sobre Moodle. La docencia en línea, además, se ha extendido al resto de niveles formativos: enseñanza presencial, cursos de másteres, expertos universitarios, extensión universitaria, etc.

El Campus Virtual de la ULPGC ofrece a muchos estudiantes la posibilidad de cursar una titulación totalmente a distancia y a otros complementar su formación presencial, con la participación de los profesores que cada vez se involucran más en fomentar la participación en los cursos virtuales.

Para facilitar su manejo, la estructura idónea y natural para cada una de las plataformas que conforman este campus es aquella que más se asemeja a la estructura tradicional de la Universidad, utilizando las mismas entidades y nomenclatura.

La información en las plataformas de enseñanza a distancia tiene que estar actualizada a diario y de manera automática, de tal forma que sea un fiel reflejo de la información académica real, permitiendo a alumnos y profesores interactuar con la plataforma en el menor lapso de tiempo posible desde que se matriculen o asignen. El sistema de información que maneja los datos de titulaciones, cursos, grupos, alumnos y profesores de la ULPGC es muy complejo y Moodle no puede trabajar directamente con él.

Por tanto, es necesario desarrollar los procedimientos necesarios para realizar esta sincronización de forma sistemática y, automáticamente, transferir la información de la base de datos corporativa de la ULPGC a la base de datos de cada plataforma virtual.

Facilitar y colaborar en la mejora de este servicio, permitiendo que todos los usuarios puedan acceder de forma automática a los cursos en los que están matriculados, es esencial para la institución y, como profesional de las TIC, ofrece un alto grado de satisfacción personal.

1.2 ESTADO ACTUAL

En el momento de iniciar el proyecto, la sincronización se realiza enviando unos ficheros de texto desde los servidores de la base de datos institucional a los servidores del Campus Virtual. En el Campus Virtual se procesan todos los registros diariamente, es decir, en cada ejecución se intenta crear cada titulación, curso y usuario, implicando un tiempo de procesamiento elevado, que, en caso de ejecutarse durante las horas de mayor uso de las plataformas, ralentizaría notablemente el rendimiento de las mismas.

Además, los ficheros de texto deben generarse íntegramente antes de cada actualización, incrementando el número de pasos y el tiempo necesarios.

El problema más crítico, desde el punto de vista operativo del Campus Virtual, radica en la mala generación o transmisión de los ficheros de texto, que podría provocar efectos indeseados como la eliminación de un usuario en una asignatura o incluso la desactivación de cuentas de usuario, sucesos ocurridos en varias ocasiones y que se han solventado utilizando copias de seguridad de días anteriores.

1.3 OBJETIVOS DEL PROYECTO

El objetivo principal del proyecto es desarrollar las extensiones de Moodle que garanticen que la información académica de la ULPGC relativa a titulaciones, asignaturas, grupos, profesores y alumnos se traslada al Campus Virtual, teniendo en cuenta las siguientes premisas:

- Automatización del proceso de sincronización: diariamente se deben sincronizar los datos de la base de datos corporativa con la base de datos del Campus Virtual sin intervención humana. Se desarrollarán los procesos necesarios para que, a diario, se sincronicen ambas bases de datos.
- No perder información: Es de vital importancia que no se pierda información accidentalmente en la plataforma virtual.
- Actualización: la información debe estar actualizada en la mayor brevedad posible. Como máximo, un día.

- **Integridad y coherencia:** la información en ambas bases de datos debe coincidir; la información en el Campus Virtual tiene que coincidir con la realidad académica de la ULPGC.
- **La información que se debe sincronizar comprende:** titulaciones, asignaturas, profesores, estudiantes, grupos y asignación de profesores y estudiantes a sus asignaturas y grupos.
- **Unidireccional:** las modificaciones realizadas en el Campus Virtual, nunca se transmitirán a la base de datos institucional.
- **Eficiencia:** el proceso de sincronización debe consumir la menor cantidad de recursos posibles, no afectando al rendimiento del Campus Virtual. En la medida de lo posible, es deseable que se pueda realizar manualmente si no se quiere esperar a la sincronización programada.

1.4 METODOLOGÍA

Este proyecto se desarrollará siguiendo el desarrollo en cascada, principalmente por dos razones:

- es la metodología empleada en el Servicio de Informática, responsable final del correcto funcionamiento de este proyecto que se desarrollará al amparo del mismo;
- se trata de un proyecto estable, donde es poco probable que surjan cambios durante el desarrollo del mismo, puesto que los requisitos están claramente definidos, y en base al estado actual se puede producir un diseño correcto, corrigiendo los problemas actuales, antes de empezar la implementación.

1.4.1 Metodología en Cascada

Es un modelo lineal de desarrollo, siguiendo un proceso secuencial, donde cada fase (análisis, diseño, implementación, pruebas, implantación y mantenimiento) se finaliza antes de pasar a la siguiente.



Figura 1-1 Fases de la metodología en Cascada

La planificación del proyecto es el elemento central de esta metodología, por lo que requiere un análisis muy detallado desde el inicio, permitiendo estimar plazos y costes con mayor precisión que otras metodologías.

Por el contrario, es poco flexible. Modificar los requisitos u objetivos del proyecto en cualquier fase del proyecto es muy complicado, de ahí la importancia del análisis inicial. Además, las pruebas y la retroalimentación tienen lugar en las últimas fases del proyecto, por lo que la capacidad de reacción es muy limitada y requiere una importante cantidad de recursos (tiempo, esfuerzo y dinero).

Brevemente, las fases de la metodología en cascada son:

1. Análisis: evaluación de los requisitos de los usuarios finales para determinar los objetivos del proyecto y disponer de una especificación completa de lo que debe hacer el sistema sin entrar en detalles internos. Señalar que todo lo que se requiere del sistema se debe definir en esta fase, no pudiéndose añadir nuevos requerimientos una vez finalizada esta fase.
2. Diseño: a partir de los requisitos, se obtiene un diseño lógico de la aplicación independiente del hardware y software a utilizar, para posteriormente transformarlo en un diseño físico en función de los sistemas hardware y software donde se implantará el producto.

3. Implementación: desarrollo del código de la aplicación en base a los requisitos y especificaciones de las fases anteriores, siendo de gran utilidad el uso de prototipos.
4. Pruebas: verificar que la aplicación funciona correctamente y cumple con las expectativas de los usuarios. El usuario final debe verificar y validar el correcto funcionamiento. Existen varios tipos de prueba, que se describirán con más detalle en el apartado correspondiente:
 - Pruebas de unidad
 - Pruebas de integración
 - Pruebas de sistema
 - Pruebas de aceptación
5. Implantación: la solución se pone en producción o se integra en otra aplicación de mayor envergadura ya en producción.
6. Mantenimiento: seguimiento de la solución en producción. El objetivo es detectar errores y solucionarlos. Las causas de estos errores pueden ser requisitos mal definidos, un diseño incorrecto o cambios en los requisitos de usuario.

1.5 PLANIFICACIÓN

A continuación se definen las fases en las que se desglosa este proyecto, incluyendo una tabla con la estimación temporal para cada una de ellas.

- Análisis de requisitos: análisis de los requisitos de los usuarios y los sistemas.
- Base de datos institucional: conocimiento básico de la base de datos desde la que se va a obtener la información a sincronizar. A groso modo, conocer dónde y cómo se almacena y cómo se puede acceder a la misma desde un sistema externo.
- Moodle: adquirir las nociones fundamentales de desarrollo en Moodle, arquitectura, convenciones, *APIs* disponibles, etc.
- Herramientas de desarrollo: familiarizarse con el software que se empleará durante el desarrollo de las extensiones.
- Diseño: diseñar las extensiones en base al análisis elaborado en la primera fase.

- Desarrollo: implementación de las extensiones a partir del diseño elaborado.
- Pruebas: antes de poner en producción, realizar pruebas del correcto funcionamiento de las extensiones y de la plataforma tras su integración.
- Implantación: instalar en las plataformas en producción las extensiones.
- Seguimiento: verificar en el Campus Virtual la validez de la sincronización empleando las extensiones implementadas. Realizar durante un tiempo prudencial que la carga y actualización de la información es congruente y coincide en ambos sistemas.
- Generación de documentación: orientada, por un lado, a los administradores de la plataforma, para el correcto uso y configuración de las extensiones, y, por otro lado, a desarrolladores que quieran modificar o personalizar las extensiones.

FASES	DURACIÓN (HORAS)
ESTUDIO	
Base de datos institucional	20
Moodle – cómo desarrollar y cargar datos externos	100
Herramientas de desarrollo	80
ANÁLISIS	
Análisis de requisitos	80
DISEÑO	
Diseño de la implementación en base al análisis realizado	80
DESARROLLO	
Desarrollo de las extensiones de Moodle	330
Pruebas	30
Implantación	40
Seguimiento	60
Generación de documentación	40
TOTAL	860

Tabla 1 Planificación

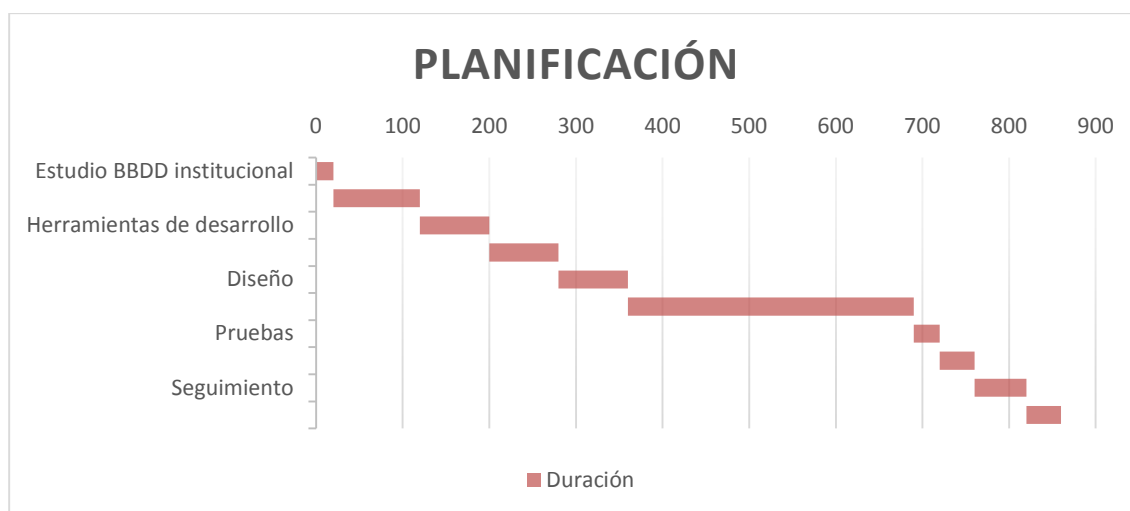


Figura 1-2 Planificación

1.6 RECURSOS NECESARIOS

Para realizar este proyecto se requiere:

- Un PC, siendo el sistema operativo indiferente.
- Un IDE, entorno de desarrollo, que facilite la programación en PHP.
- Conexión a internet.
- Acceso a una instalación de Moodle similar a las del Campus Virtual de la ULPGC. En esta instancia de Moodle se desarrollará el código. Si no se dispone de la misma, son necesarios un servidor Apache y un servidor MySQL para instalar Moodle. Desde el servidor web es necesario conectarse a la base de datos institucional para obtener la información a sincronizar.

Las diferentes herramientas y tecnologías de desarrollo a emplear se describen en el capítulo **5 Herramientas y tecnologías**.

1.7 ESTRUCTURA DE LA MEMORIA

Esta memoria arranca poniendo en contexto al lector mediante una concisa introducción al ámbito y entorno tecnológico en el que se desarrolla este proyecto.

- **Capítulo 1. Introducción:** presentación del estado actual y la problemática existente, justificando la realización de este proyecto y planteando cómo se va a llevar a cabo.

- **Capítulo 2. E-learning:** definición de *e-learning*, los sistemas de gestión de aprendizaje (*LMS*, por sus siglas en inglés, *Learning Management System*) y la historia y presente del *e-learning* en la ULPGC.
- **Capítulo 3. Moodle:** introducción a Moodle, organización y conceptos básicos.
- **Capítulo 4. Sistema de información de la ULPGC:** nociones de los elementos del sistema de información de la ULPGC que entran en juego en el desarrollo de este proyecto.

El siguiente bloque de capítulos se centran en el desarrollo del proyecto, desde el análisis hasta la realización de las pruebas, siguiendo la metodología en cascada descrita anteriormente en este capítulo. Adicionalmente incluye un capítulo que contempla las especificidades del desarrollo en Moodle y otro recogiendo el resultado de la puesta en producción y posibles líneas de trabajo futuras.

- **Capítulo 5. Herramientas y tecnologías:** descripción de las herramientas y tecnologías empleadas.
- **Capítulo 6. Desarrollo en Moodle:** introducción al desarrollo en Moodle.
- **Capítulo 7. Análisis:** especificación de los requisitos de usuario y estudio inicial del sistema a desarrollar.
- **Capítulo 8. Diseño:** diseño de la solución a partir del análisis realizado.
- **Capítulo 9. Implementación del sistema:**
- **Capítulo 10. Desarrollo de las pruebas:** detalle de la realización de las pruebas diseñadas previamente y los resultados obtenidos.
- **Capítulo 11. Conclusiones y trabajo futuro:** impresiones de los resultados obtenidos tras la puesta en producción, así como mejoras y nuevas funcionalidades a desarrollar.

En la bibliografía y los anexos se recoge material de referencia para quien desee profundizar más en el trabajo realizado, así como los manuales para la instalación y configuración del sistema de sincronización en un entorno real.

- **Bibliografía:** referencias bibliográficas utilizadas, tanto para el desarrollo como para la redacción de este documento.
- **Anexo I. Instalación de Moodle:** pasos a seguir para instalar Moodle.

- **Anexo II. Subversion:** uso básico de Subversion para integrar el desarrollo en el mecanismo de control de versiones del Servicio de Informática de la ULPGC.
- **Anexo III. Instalación y configuración de Eclipse:** preparación de este IDE para optimizar el desarrollo de código en Moodle.
- **Anexo IV. Guía de estilo de código:** resumen de las directrices que marca el equipo de desarrollo de Moodle para facilitar la integración y mantenimiento de código desarrollado para esta plataforma.
- **Anexo V. Guía de seguridad:** consideraciones a tener en cuenta para garantizar la seguridad del código desarrollo para Moodle.
- **Anexo VI. Base de datos: guía de estilo y XMLDB:** directrices para la creación de estructuras de base de datos en Moodle.
- **Anexo VII. Manuales del sistema:** manuales de instalación y configuración del sistema de sincronización, dirigidas principalmente a los administradores de sistemas y de administradores de la plataforma.

En el CD adjunto está disponible el código fuente así como una copia de esta memoria. El CD está organizado como sigue:

- **codigo_fuente:** código fuente del sistema de sincronización. Basta con copiar el contenido en el directorio raíz de una instalación de Moodle.
 - **auht/casulpgc:** extensión de identificación, sincronización de usuarios.
 - **enrol/sinculpgc:** extensión de matriculación, sincronización de matrículas.
 - **local/sinculpgc:** extensión local, sincronización de centros, categorías, cursos, grupos y asignaciones de grupo.
- **memoria:** esta memoria en formato PDF.

2 E-LEARNING

El e-learning es el resultado de aplicar las nuevas tecnologías en el ámbito de la formación y el aprendizaje. Se está extendiendo de tal forma, tanto a nivel empresarial como educativo, sobre todo universitario, que no se concibe una universidad sin un Campus Virtual o formación en línea, lo que hace posible tanto el apoyo a la enseñanza presencial como la impartición de titulaciones y la obtención de los correspondientes títulos.

Hay muchos términos que se usan indistintamente para referirse a la enseñanza en línea que no significan exactamente lo mismo. Se ha generalizado el uso de *e-learning* para definir el aprendizaje en línea, pero tal como se definió en el párrafo anterior, el *e-learning* engloba cualquier tipo de enseñanza que implique el uso de las TIC, sin implicar la educación a distancia ni el uso de internet. Por otro lado, la enseñanza a distancia no conlleva el uso de las TIC, ya que puede reducirse al uso de material impreso por parte del alumno, consultas a un tutor puntualmente y la realización de una prueba para evaluar los conocimientos adquiridos. Una combinación de ambas aproximaciones conforma el *e-learning on-line*.

Para facilitar la lectura de esta memoria, utilizaremos indistintamente los términos *on-line*, en línea y e-learning, teniendo siempre presente que nos referiremos al *e-learning online*, salvo que se indique explícitamente.

2.1 E-LEARNING ON-LINE

Las crecientes necesidades formativas de la sociedad requieren de nuevas fórmulas de enseñanza, que se adapten a la demanda de la comunidad educativa. La posibilidad de estudiar cuándo, dónde y cómo se desee se convierte en una estrategia formativa que utiliza Internet o una Intranet como tecnología de distribución de la información. De la unión de la enseñanza con las nuevas tecnologías surge el e-learning.

Los principales beneficios de esta modalidad de enseñanza son:

- Reducción de costes: con unos costes fijos, los correspondientes a la infraestructura, permite formar a más alumnos y ofrecer mayor cantidad de estudios y recursos.
- Evita desplazamientos: ni alumnos ni profesores tienen que desplazarse, pudiendo realizar su actividad diaria desde cualquier punto en el que dispongan de conexión a

internet. Incluye un ahorro de costes y reducción en el impacto medioambiental que genera el uso de vehículos.

- Facilita la comunicación y la interactividad: hay muchas herramientas que facilitan la comunicación tanto horizontal (entre compañeros) como vertical (con los profesores). Foros, chats, mensajería, son ya parte de cualquier entorno colaborativo y cada vez es más frecuente el uso de videoconferencia como instrumento para comunicarse con uno o varios participantes de un curso.
- Flexibilidad: los estudiantes pueden marcar su propio ritmo. Idealmente se podría acceder a los recursos las 24 horas del día los 7 días de la semana (24x7x365).
- Combinación de materiales educativos: las limitaciones en el uso de las pizarras o transparencias se eliminan con un amplio abanico de herramientas multimedia que posibilitan crear contenido que enriquecen la experiencia del usuario, combinando audio, vídeo e imágenes, potenciando la interacción del usuario.

Además de las ventajas mencionadas, hay otras serie de características relevantes asociadas al *e-learning online* :

- Vivo: contenidos y actividades actualizados, con variedad de fuentes y facilidad para acceder y modificar los mismos.
- Personalizado: cada usuario debe poder obtener formación y/o material acorde a sus necesidades, estando disponible cuando el usuario puede o quiere acceder.
- Colaborativo: la colaboración facilita e impulsa el aprendizaje. Son esenciales los mecanismos que permiten interactuar a los usuarios.
- Individualizado: cada usuario debe poder elegir cómo adquirir los conocimientos, utilizar los recursos que mejor se adapten a su metodología de estudio.

El *e-learning on-line* se fundamenta, tecnológicamente, en los Sistemas de Gestión de Aprendizaje (*LMS*), entornos web desde los que se ofrecen, administran y gestionan tanto los contenidos formativos como la actividad de los estudiantes.

2.2 SISTEMAS GESTIÓN DE APRENDIZAJE (LMS)

Los Sistemas de Gestión de Aprendizaje (*Learning Management Systems*, *LMS*) o plataformas de aprendizaje, son un software instalado generalmente en un servidor web, que se emplea para

crear, aprobar, administrar, almacenar, distribuir y gestionar las actividades de formación virtual. Puede utilizarse como complemento a la enseñanza presencial o a la enseñanza a distancia tradicional.

Para simplificar, utilizaremos las siglas LMS para referirnos a este tipo de software.

Los LMS se centran en gestionar los contenidos y la interacción de los usuarios. La creación de los contenidos se realiza utilizando Sistemas de Gestión de Contenidos para el Aprendizaje (LCMS, siglas de *Learning Content Management Systems*) o herramientas de autor, generalmente externas al LMS. Cada vez es más frecuente que los LMS incluyan herramientas para generar los propios contenidos.

2.2.1 Usuarios

En un LMS podemos encontrar diferentes tipos de usuarios:

- Administradores: instalan, configuran y gestionan el buen funcionamiento del LMS. No intervienen en la formación.
- Diseñadores instruccionales: comúnmente conocidos como creadores o diseñadores de cursos, que definen la estructura de los cursos y utilizan los contenidos para estructurar los mismos. En muchos casos son los propios profesores.
- Profesores: utilizan los contenidos del curso para formar a los alumnos o complementar la docencia presencial. Adicionalmente, en muchos LMS disponen de funcionalidades para evaluar y hacer un seguimiento del aprendizaje de los alumnos.
- Alumnos: acceden al LMS para adquirir conocimientos, interactuar con otros usuarios y desarrollar y/o entregar diferentes actividades propuestas por los profesores de los cursos.

2.2.2 Características

Según Clarenc (Clarenc, 2013), las características que debería cumplir un LMS son:

- Interactividad: los LMS deben disponer de las herramientas necesarias para ofrecer suficiente interactividad, de tal forma que los usuarios puedan comunicarse e intercambiar conocimientos, sin dejar de ser los protagonistas de su propio aprendizaje.
- Flexibilidad: las plataformas no deben ser rígidas a la hora de implementar un plan de estudio o diseñar un curso. Deben permitir modificar la estructura general o la

configuración de un curso, adaptándose a la organización de un centro o al método pedagógico que decidan emplear los profesores.

- Escalabilidad: un LMS debe ofrecer un rendimiento aceptable a los usuarios, independientemente del número de usuarios registrados y/o activos.
- Estandarización: tiene dos vertientes. Por un lado que su uso sea acorde a las convenciones con las que están familiarizados los usuarios al utilizar otras aplicaciones. Por otro lado, que permita integrar con facilidad materiales que hayan sido elaborados con otras herramientas.
- Usabilidad: rapidez y facilidad con la que los usuarios pueden realizar sus tareas dentro del LMS. Contempla aspectos de efectividad, eficiencia y accesibilidad, que generan el grado de satisfacción del usuario.
- Funcionalidad: debe cumplir con los requerimientos y necesidades de los diferentes usuarios, permitiéndoles desempeñar su labor. En el siguiente apartado se describen las principales funciones que debe cumplir.
- Ubicuidad: es la capacidad de una plataforma de hacer sentir al usuario presente en todo momento, independientemente de dónde esté o cómo se conecte, y de que encontrará en ella todo aquello que necesite para avanzar en sus estudios.
- La integración y articulación de otras cuatro características, funcionalidad, usabilidad, ubicuidad e interactividad. Es la capacidad que tiene una plataforma de fidelizar a un usuario a través de uso, ofreciéndole una experiencia satisfactoria que le motive a continuar utilizándola.

2.2.3 Funciones

Las principales funciones que debe ofrecer un LMS moderno son:

- Gestión de usuarios: permitir la creación de usuarios, modificación de sus datos y asignación a cursos con diferentes roles (profesor, alumno, etc.).
- Gestión de contenidos: facilitar la adición de contenido (material didáctico) y creación de actividades a realizar por los alumnos.
- Evaluaciones: mecanismos para valorar la adquisición de conocimiento por parte de los alumnos.

- Seguimiento del proceso de aprendizaje: mecanismos para determinar dónde se encuentra un estudiante en cada momento y ofrecerle un aprendizaje a medida. Útil para identificar dificultades durante el aprendizaje.
- Elaborar informes: los informes son una poderosa herramienta para evaluar el desempeño tanto a nivel individual como colectivo de los usuarios, limitándose a un curso o a toda una plataforma.
- Ofrecer herramientas de comunicación: la interacción entre profesores y alumnos es imprescindible; la colaboración entre alumnos mejora exponencialmente el aprendizaje. Son necesarias herramientas de comunicación uno a uno, uno a muchos y muchos a muchos (foros, chats, mensajería privada, etc.).

Observar que no se contempla la elaboración de contenidos entre las funcionalidades que debe ofrecer un LMS, sino la gestión de los mismos. Los entornos destinados a la creación de contenidos se conocen como Sistemas de Gestión de Contenido para Aprendizaje (LCMS, siglas de *Learning Content Management Systems*).

La información de cursos y usuarios normalmente se almacena en los sistemas de información de las organizaciones, siendo la integración de éstos con el LMS un factor determinante a la hora de implantar una plataforma de enseñanza *on-line*.

2.3 E-LEARNING EN LA ULPGC

En el año 1994, el CICEI (Centro Informático y de Comunicaciones del Edificio de Ingeniería) asume la implantación e integración de las Tecnologías de la Información en el ámbito de la ULPGC.

En 1997 en el CICEI se comienza a trabajar con la versión beta de WebCT. Este mismo año se inicia un proyecto de innovación docente, denominado *Proyecto INNOVA*, en el que participaron más de 30 profesores de diferentes departamentos de la Universidad. La plataforma utilizada, IVA (Interfaz Virtual de Aula), fue un entorno desarrollado por el CICEI sobre la primera versión de WebCT.

La primera titulación en línea de la ULPGC, Psicopedagogía en línea, se ofrece en febrero del año 2002, utilizando IVA. Esta primera edición se impartió en modalidad semipresencial, contando con sesiones presenciales, material impreso, tutorías y consultas en línea. En esta edición se matricularon 90 estudiantes residentes en cuatro de las islas. En el curso 2003/04 el

número de matriculados aumenta a 185, con presencia de alumnos no residentes en las Islas Canarias, y en el curso 2004/05 ya superan los 200.

En el curso 2003-04 se desarrolló en la propia ULPGC una aplicación como soporte a la enseñanza presencial, no relacionada ni con IVA ni con WebCT. Se bautizó como EVA o LPE (La Plataforma Educativa). Aunque resultó excesivamente no funcional, fue la insatisfacción con la misma la que causó la búsqueda de otra plataforma mejor y descartar desarrollos a medida.

A partir de los resultados obtenidos en el uso de LPE, se definieron una serie de criterios básicos para la evaluación de un LMS como alternativa a la misma:

- Basado en web y multiplataforma
- Gestión de acceso unificada y permiso basado en roles
- Flexibilidad en el uso de actividades y módulos docentes
- Fácil diseño y creación de cursos
- Trabajo colaborativo
- Seguimiento de la actividad del alumno
- Características técnicas:
 - Seguridad
 - Facilidad para añadir/desarrollar nuevas funcionalidades
 - Escalabilidad frente al aumento en el número de usuarios
 - Integración con otros sistemas

Se realizó una comparativa entre más de 50 plataformas de e-learning de libre distribución, dado que el presupuesto era nulo. En función de los criterios anteriormente expuestos y otra serie de características deseables en este tipo de entornos, se concluyó que Moodle destacaba claramente por su interfaz amigable, siendo la herramienta más fácil de manejar, su extensa interoperabilidad, el gran número de módulos de actividades, sus características avanzadas de trabajo colaborativo, siendo enormemente potente y flexible.

En el año 2004 se toma la decisión de adoptar Moodle como plataforma de e-learning, desarrollado como software libre por un equipo liderado por Martin Dougiamas, antiguo

desarrollador de WebCT. Se inicia la docencia de Másteres y Expertos, programas de Doctorado, cursos de Extensión Universitaria y de Formación Continua.

Bajo el proyecto de desarrollo del Campus Virtual de la ULPGC, se pone en marcha la plataforma Apoyo a la Enseñanza Universitaria Presencial, en la que se aprovecha la experiencia obtenida en la formación no presencial para mejorar la calidad de la modalidad de enseñanza tradicional, y la plataforma Espacios Virtuales de Trabajo, en la que se ofrece toda la potencia de Moodle para la estructuración de todo tipo de grupos de trabajo a distancia en docencia, investigación y gestión.

Fruto de la estrecha colaboración de la ULPGC con la comunidad mundial de Moodle, en 2005 se organiza en la ULPGC un congreso de desarrolladores de Moodle (MoodleMoot'05).

En el curso académico 2005/06 se inician dos nuevas titulaciones de grado online: Maestro en Educación Primaria y Diplomatura en Turismo. En el curso siguiente, 2006/07, se añaden dos nuevas titulaciones: Diplomatura en Trabajo Social y Diplomatura de Relaciones Laborales. Paralelamente, en la plataforma Apoyo a la Enseñanza Presencial, son cada vez más los profesores y estudiantes que participan, de tal forma que todas las titulaciones y asignaturas en modalidad presencial tienen su espacio virtual.

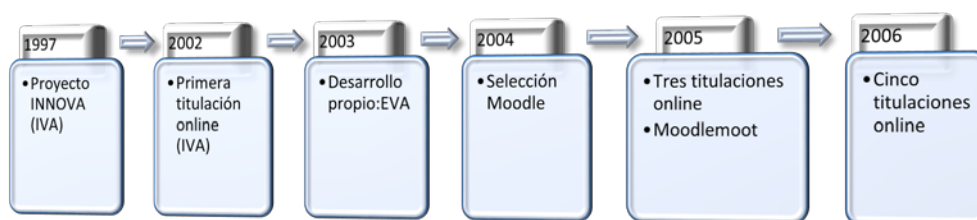


Figura 2-1 Principales hitos e-learning en la ULPGC

La ULPGC, en el curso académico 2014-15, vigente durante la realización del proyecto, mantiene una plataforma de e-learning destinada a la enseñanza presencial, en la que todas las titulaciones y asignaturas están presentes, así como una plataforma destinada exclusivamente a las titulaciones en línea. En el siguiente apartado se detalla la organización del Campus Virtual de la ULPGC.

2.4 ORGANIZACIÓN DEL CAMPUS VIRTUAL DE LA ULPGC

Un Campus Virtual, análogamente a los campus universitarios, es el espacio físico, virtual, administrativo, tecnológico y educativo donde se desarrolla toda la experiencia institucional, y se ha consolidado como el pilar del e-learning en las universidades. Uno de los principales elementos de un Campus Virtual es el LMS.

El Campus Virtual de la ULPGC lo componen varias instancias de Moodle independientes, cada una con su propia instalación y base de datos:

- **Teleformación:** se encarga de las titulaciones impartidas exclusivamente en línea, a distancia, por la Estructura de Teleformación.
- **Grado y Posgrado:** da servicio a titulaciones oficiales de grado y másteres y doctorados de posgrado (1^{er}, 2^o y 3^{er} ciclo), ya sean impartidas en régimen presencial, semipresencial o a distancia.
- **Trabajo colaborativo:** ofrece espacios genéricos de colaboración para grupos de trabajo o de investigación de la ULPGC o externos.
- **Campus social:** ofrece espacio a cursos de biblioteca, centros, foro de calidad y evaluación, punto de encuentro y sala de profesores.
- **Otras Enseñanzas:** destinada a estudios de másteres y expertos, cursos de Extensión Universitaria, programas formativos especiales (Diploma de Estudios Canarios, Diplomas de Estudios Europeos, Peritia et Doctrina). También aloja cursos externos a la ULPGC, mediante convenios de colaboración.

Plataforma	Teleformación	Grado y Posgrado	Trabajo colaborativo	Campus social	Otras Enseñanzas
Cursos	504	4822	465	63	1225
Usuarios	2338	21920	4269	50994	7747

Tabla 2 Plataformas del Campus Virtual de la ULPGC

3 MOODLE

La palabra Moodle es el acrónimo, en inglés, de *Modular Object-Oriented Dynamic Learning Environment*, cuya traducción es Entorno de Aprendizaje Dinámico Modular Orientado a Objetos.

Es una aplicación que se puede categorizar como un gestor de contenidos educativos (LMS, *Learning Management System*), subcategoría a su vez de los gestores de contenidos (CMS, *Content Management System*).

En pocas palabras, ofrece un espacio donde un centro educativo o una organización gestionan recursos educativos proporcionados por unos docentes y organiza el acceso a los mismos por parte de los alumnos, facilitando la comunicación entre ambos grupos.

La primera versión de Moodle apareció el 20 de agosto de 2002; desde entonces han aparecido nuevas versiones de forma regular que han ido incorporando nuevos recursos, actividades y mejoras demandadas por la comunidad de usuarios. Moodle es un software de código abierto, bajo licencia GPL, escrito en PHP.

A la redacción de esta memoria, Moodle es la plataforma LMS más popular del mundo y está traducido a 75 idiomas e incluye más de 27.000 sitios registrados.

La estructura de Moodle gira en torno a los cursos, usuarios y roles, tres conceptos interrelacionados y dependientes entre sí (Büchner, 2011). Definiremos estos tres conceptos y como se relacionan, así como otros de menor importancia pero igualmente relevantes para la comprensión de este proyecto.

3.1 ORGANIZACIÓN INTERNA DE MOODLE

Para tener una visión de conjunto de la importancia de los cursos, usuarios y roles, se utilizará este diagrama donde se resalta la importancia de los tres conceptos y como se relacionan con otros elementos.

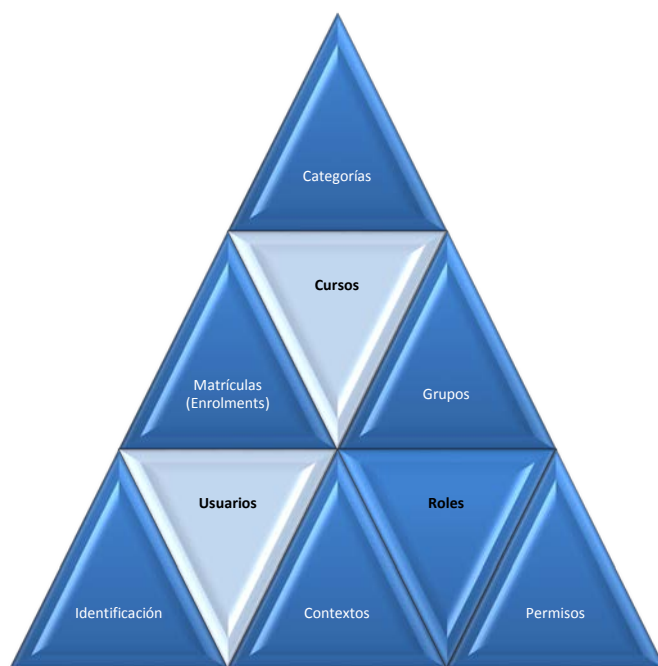


Figura 3-1 Estructura de Moodle

Haciendo un recorrido siguiendo el sentido de las agujas del reloj y partiendo de la esquina inferior izquierda:

- Los usuarios tienen que superar un proceso de *Identificación* para obtener acceso a Moodle.
- Luego tienen que disponer de *Matrícula* en aquellos *Cursos* en los que quieran participar, estando los cursos organizados en *Categorías*.
- Los *Grupos* y *Cohortes* permiten agrupar usuarios a nivel de curso o de sitio respectivamente.
- Los usuarios obtienen *Roles* en diferentes *Contextos*, dependiendo qué puede hacer cada rol únicamente por los *Permisos* que se le han asignado al mismo.

Actividades y recursos

Una **actividad** es un elemento que permite a los usuarios interactuar con otros usuarios o con el profesor. Ejemplos de actividades es un foro, enviar un documento respondiendo a un enunciado o responder un cuestionario.

Un **recurso** es un elemento que un profesor añade a un curso como apoyo al aprendizaje, como puede ser un archivo, un vídeo o un enlace a una página web. La principal diferencia con las

actividades es que los recursos son estáticos. Los estudiantes solo pueden verlo o leerlo, pero no interactuar con el mismo.

Cursos

Los cursos son el elemento central de Moodle ya que son los contenedores donde el proceso de enseñanza tiene lugar. Es el espacio donde los profesores añaden el material docente, en forma de recursos, crean actividades, resuelven dudas y evalúan el trabajo de los alumnos. Por su parte, los alumnos leen, escuchan o visualizan los materiales docentes, participan en actividades, envían sus trabajos y colaboran con otros.

La organización interna puede variar, pero normalmente incluyen una serie de **secciones** donde se muestra el material y **bloques** laterales ofreciendo características o información adicionales.

Categorías

Los cursos se organizan jerárquicamente en **categorías**, que son únicamente contenedores de cursos, de forma similar a como se organizan los ficheros en un sistema operativo. Pueden contener subcategorías, que a su vez pueden tener otras subcategorías y así sucesivamente. La organización de los ficheros en un disco duro es un símil: las categorías serían como carpetas y los cursos los ficheros.

La estructura jerárquica de los elementos hasta ahora mencionados:

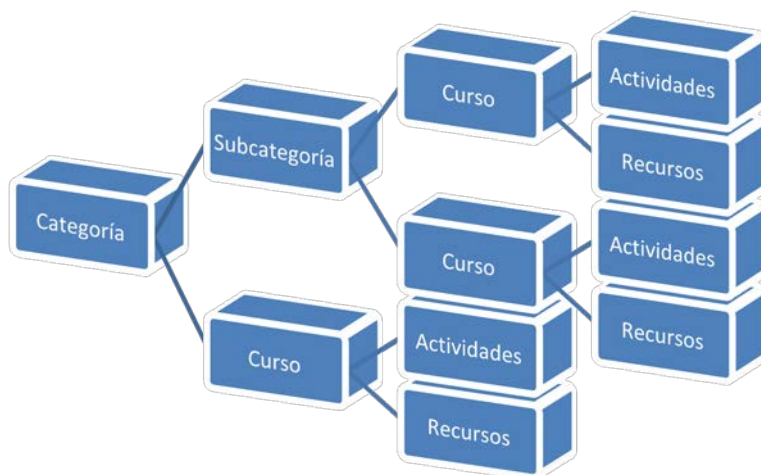


Figura 3-2 Jerarquía de categorías y cursos

Un curso siempre pertenece a una única categoría. Ni puede pertenecer a más ni no pertenecer a ninguna. La única excepción es la página principal, que internamente se considera un curso que no pertenece a ninguna categoría y que no puede eliminarse.

Usuarios

Cualquiera que pueda acceder al sistema es un usuario. Es importante diferenciar entre identificación y matrícula.

Los usuarios en Moodle tienen que identificarse para poder acceder al sistema. La identificación garantiza el acceso de los usuarios al sistema facilitando un usuario y una contraseña. Moodle soporta gran cantidad de mecanismos de identificación, siendo CAS el utilizado en la ULPGC.

Por su parte, la matrícula tiene lugar a nivel de curso, pero el usuario tiene que estar identificado antes de poder matricularse en un curso.

La siguiente figura ilustra la diferencia entre identificación y matrícula:

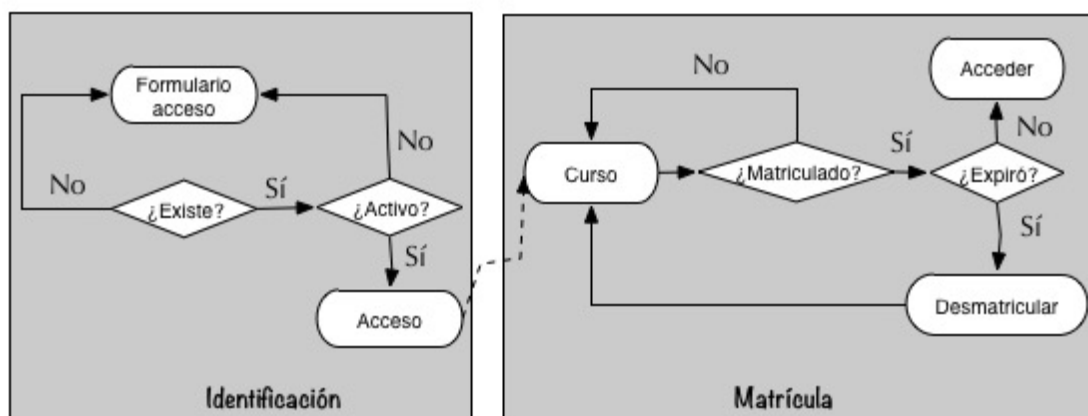


Figura 3-3 Diferencias entre identificación y matriculación

Roles

Hay varias categorías o roles, que definen un conjunto de permisos que determinan qué puede y qué no puede hacer un usuario en el contexto al que se le aplica. Cada rol está asociado a un ámbito, denominado contexto, y está definido por un conjunto de permisos (capacidades). Por ejemplo, un profesor puede calificar una tarea mientras que un estudiante no puede. Sin embargo, un estudiante puede enviar una tarea para su corrección y un profesor no. Ejemplo de roles que vienen definidos por defecto son los de Administrador, Profesor y Estudiante.

Un contexto es un área acotada de Moodle en la que se puede asignar roles a usuarios. A un mismo usuario se le pueden asignar diferentes roles en diferentes contextos, donde un contexto puede ser un curso, una categoría, un módulo de actividad, un usuario, un bloque, la página de inicio o todo Moodle. Por ejemplo, a un usuario se le puede asignar el rol Administrador para

todo el sistema y, adicionalmente, el rol Profesor en un curso del que es responsable. O a un estudiante se le podría asignar el rol Profesor para que ejerza de moderador en un foro.

Grupos

A nivel de curso, se puede asignar a uno o varios usuarios a uno o más grupos. Dentro del curso se puede asignar una actividad o un recurso a un grupo, de tal forma que solo los miembros de ese grupo puedan acceder a la actividad o recurso.

Por lo tanto, un posible flujo de trabajo sería el siguiente:

1. Se crean los cursos y categorías.
2. Se crean los grupos.
3. Se crean los usuarios.
4. Se matriculan los usuarios en los cursos y se asignan roles.
5. Se asignan los usuarios a los grupos.

4 SISTEMA DE INFORMACIÓN DE LA ULPGC

A continuación se detallan aquellos elementos del sistema de información de la ULPGC dentro del ámbito de la realización de este proyecto.

4.1 ORÍGENES

A finales de los años noventa se plantea la necesidad de realizar un profundo cambio en las aplicaciones de gestión debido principalmente, a dos motivos:

- Obsolescencia del sistema de programación y entornos de ejecución
- Introducción del euro como moneda oficial

Para renovar las aplicaciones de gestión se decide tomar ORACLE como sistema gestor de bases de datos.

A diferencia de la aplicación antigua, en la que todos los objetos de la base de datos pertenecían a un propietario único, en la nueva aplicación se han definido esquemas y usuarios para los diferentes servicios y entidades dentro de la ULPGC, de manera que cada objeto de la base de datos pertenece a un propietario concreto. Para acceder a cada objeto de forma independiente del usuario se crean sinónimos públicos. De esta forma, existen un usuario y un esquema para el servicio de Gestión Académica y otro usuario y esquema para el Vicerrectorado de Ordenación Académica y EEES.

El Campus Virtual de la ULPGC pretende ser un reflejo de la actividad docente y académica, con la misma estructura y planificación. Consecuentemente, y a más bajo nivel, la base de datos de Moodle tiene que contener la misma información que la base de datos corporativa, adaptándola en forma y contenido. Esta es la razón que hace necesario un mecanismo que sincronice la base de datos corporativa en Oracle, núcleo del sistema de información de la ULPGC, con la base de datos de Moodle en MySQL. No es posible hacer un trasvase de información lineal, puesto que las estructuras de ambas bases de datos son completamente diferentes.

La información docente y académica se almacena principalmente en dos esquemas, mencionados anteriormente: el esquema del servicio de Gestión Académica y el esquema del

Vicerrectorado de Ordenación Académica y EEES. En el apartado 4.3 se define qué es un esquema.

4.1.1 Servicio de Gestión Académica

La misión del servicio de Gestión Académica es gestionar la planificación administrativa necesaria desde que los alumnos realizan la PAU o la prueba de acceso que corresponda hasta que finalizan sus estudios correspondientes a titulaciones de 1º y 2º ciclo y de grado, gestionar las matrículas y preinscripciones, las becas, las ayudas, los programas formativos especiales, las actividades de extensión universitaria de cultura y de acción social.

La información relativa a matrículas, datos personales de alumnos, asignación de grupos de docencia, titulaciones que se imparten se almacena en este esquema.

4.1.2 Vicerrectorado de Ordenación Académica y EEES

Entre las competencias del Vicerrector de Ordenación Académica y EEES, encargada de la ordenación y planificación académica, se incluyen:

- Ordenación y seguimiento de las titulaciones de la ULPGC:
- Programación y seguimiento de la libre configuración en la Universidad.
- Programar y, dentro de sus competencias, llevar a cabo todos los trámites necesarios para la implantación de nuevas titulaciones en la Universidad, así como de los planes de estudios y todos los asuntos relacionados con la adaptación de los estudiantes entre diferentes planes de estudios.
- Establecimiento de relaciones con los centros adscritos, atendiendo a la ordenación y seguimiento de su actividad.
- Planificación, organización, seguimiento y difusión de las actividades relativas a la formación de posgrado.

Tendremos que acceder a este esquema para obtener la asignación de docentes a cursos, conocer las vinculaciones entre diferentes asignaturas.

4.2 SERVICIO DE IDENTIFICACIÓN CENTRALIZADA

La web institucional de la ULPGC ofrece contenidos de interés tanto para toda la comunidad universitaria como cualquier persona interesada en conocer los estudios que se imparten y los servicios que ofrece. Adicionalmente, se puede acceder a información de carácter personal así como realizar diversos trámites administrativos y académicos que requieren autenticación por parte del usuario.

Para facilitar la gestión y el acceso de los usuarios, la ULPGC cuenta con un sistema centralizado de autenticación, basado en el proyecto *Central Authentication Service (CAS)*. Este sistema proporciona *Single Sign On*, usa protocolos abiertos bien documentados, sus componentes son de fuente abierta, dispone de clientes para PHP, Drupal y Moodle entre otros, y está soportado por una comunidad de desarrolladores activa y extensa.

Con la implantación de este nuevo sistema, mejora la seguridad de los usuarios y su experiencia de navegación, ya que pueden acceder a todos los servicios con una sola autenticación

Siguiendo esta línea, la autenticación en el Servicio de Identificación Centralizada de la ULPGC será requisito para acceder al Campus Virtual.

4.3 BASE DE DATOS INSTITUCIONAL: ORACLE

Un **esquema** en Oracle es el conjunto de todos los objetos (tablas, paquetes, vistas, secuencias, etc.) propiedad de una cuenta de usuario. Aunque no son exactamente lo mismo, se suele usar indistintamente los términos esquema y usuario.

La base de datos institucional está dividida en esquemas, algunos de ellos de uso común y otros asociados a un área de gestión o servicio de la ULPGC. Así tenemos un esquema para el servicio de Gestión Académica, otro para Vicerrectorado de Ordenación Académica, otro para el departamento de Contabilidad, etc. La información que se debe cargar en el Campus Virtual se encuentra en diferentes esquemas y, dentro de cada esquema, en múltiples tablas y/o vistas, por lo que no es posible realizar una relación uno a uno que permita poblar las tablas del Campus Virtual desde las tablas de la base de datos institucional.

Fue necesario crear un esquema intermedio en la base de datos institucional, similar a un *Data Mart*. Un *Data Mart*, sin entrar en detalles, es un subconjunto de datos tratados y extraídos de una base de datos, organizados para facilitar su explotación por parte de otro sistema o área de negocio.

No se entrará en detalles del desarrollo de este *Data Mart*, desarrollado por el autor de este proyecto, por estar fuera del ámbito de esta memoria, pero se ofrece un esbozo de la estructura del mismo, necesario para tener claro desde dónde se obtiene la información a sincronizar en el Campus Virtual.

Para más detalles consultar el apartado **7.3.1 Sistema de Información de la ULPGC**.

4.4 ARQUITECTURA FÍSICA DEL CAMPUS VIRTUAL DE LA ULPGC

En el momento de realizar el proyecto, está instalada la versión 2.3.1+ de Moodle, con las modificaciones necesarias para adaptarla a las necesidades de la ULPGC.

4.4.1 Conexión Internet e Intranet

La conexión principal de la ULPGC con Internet se realiza a través de una línea de 622 Mbps con RedIris. Se dispone de una línea de respaldo 100 Mbps, también con RedIris. Estas líneas permiten mantener la adecuada capacidad de conexión para atender la demanda de tráfico de las más de 25.000 personas (estudiantes y profesores) que usan el Campus Virtual. El acceso a las diferentes sedes de la ULPGC se realiza a través de la red ULPnet.

Esta red es de vital importancia no sólo para satisfacer las necesidades de profesores e investigadores y PAS de cada Centro, sino también para asegurar la conectividad en las aulas de docencia presencial y, especialmente, de las Aulas de Informática a disposición de los estudiantes (no ocupadas en docencia de asignaturas específicas). La ULPGC cuenta con al menos un Aula de Informática de libre disposición en todos sus edificios, así como Aulas de Informática abiertas permanentemente, incluso por la noche. Esta oferta es importante no tanto en la docencia en régimen de Teleformación propiamente dicho (donde es asumido que los estudiantes remotos disponen de sus propios equipos informáticos) sino para el uso del Campus Virtual como Apoyo a la Enseñanza Presencial. La disponibilidad de Aulas de Informática de libre disposición por parte de los alumnos es esencial para que las posibilidades de aprendizaje autónomo y gestión personal del tiempo, en línea con las directrices del EEES, que permite el Campus Virtual puedan ser aprovechadas y materializadas y no quede en una herramienta más de apoyo a la clase magistral en el aula.

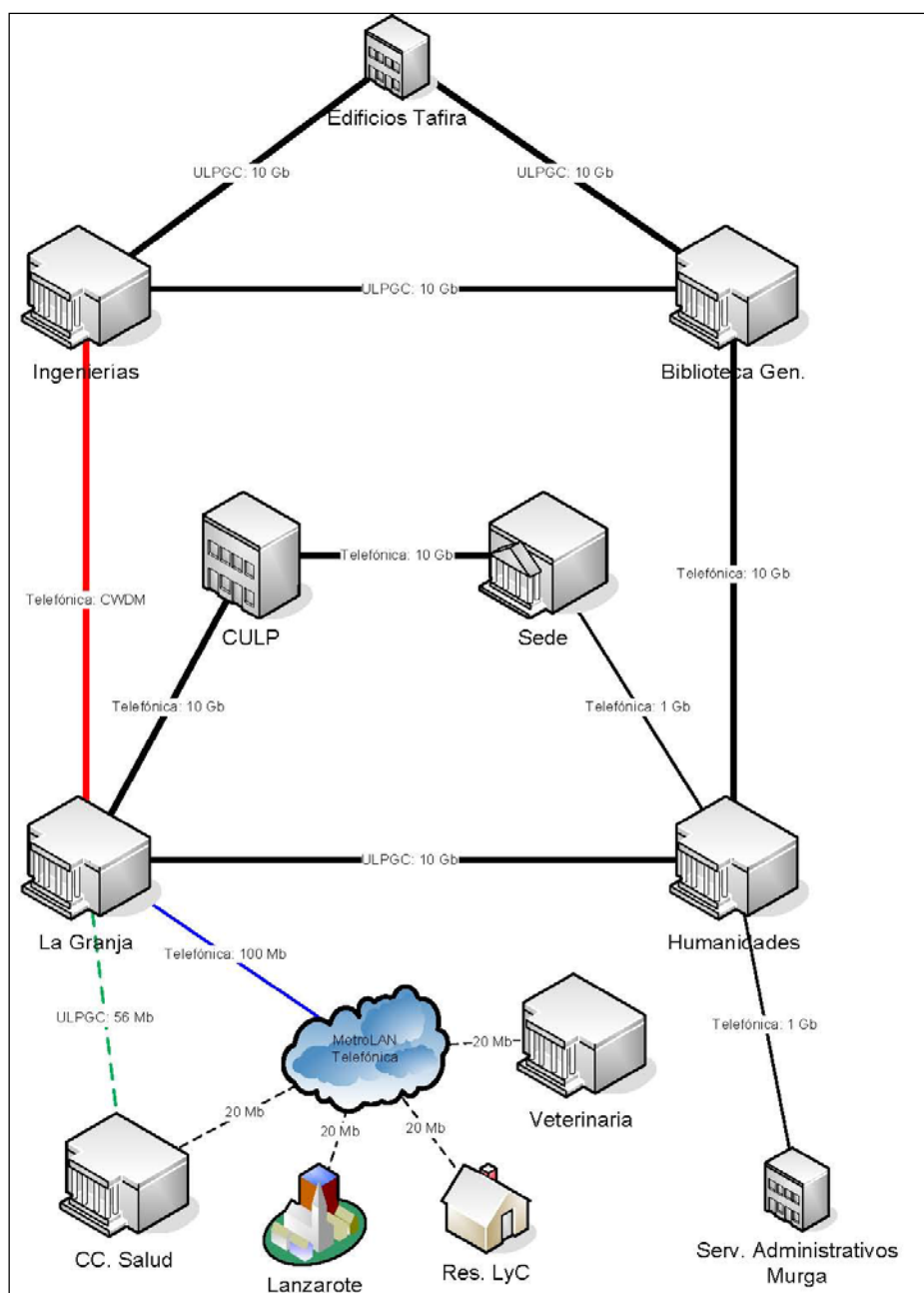


Figura 4-1 Esquema ULPnet

4.4.2 Servidores

El Campus Virtual está montado en una serie de servidores virtuales, duplicados y dotados de balanceadores de carga para garantizar la alta disponibilidad, que se ejecutan sobre un clúster VmWare y se almacenan los datos en un clúster de base de datos MySQL

Los servidores están protegidos por cortafuegos redundantes, se realiza copia de seguridad diaria mediante el software Networker y se mantiene una monitorización constante del rendimiento y de la disponibilidad mediante la herramienta Zabbix.

Además, utiliza varios recursos IT adicionales del Servicio de Informática. Así, el acceso se realiza a través de la página Web institucional y el registro y autenticación de los usuarios de todas las plataformas está centralizado en un servidor LDAP. Los estudiantes disponen para su uso en Moodle de una cuenta de correo en un servidor IMAP.

Los cuatro servidores físicos tienen las siguientes características: HP Proliant DL580 G5, con 4 procesadores Intel Xeon E7310 @ 1.60Ghz quad-core, con 32 Gb RAM

Cada una de las instalaciones mencionadas anteriormente dispone de dos frontales web, con la excepción de la plataforma de apoyo a la enseñanza presencial (con 12 servidores virtuales) y teleformación (4 servidores).

En cuanto al backend de bases de datos, es proporcionado por dos servidores en los que hay instalado un clúster MySQL versión 5.

Por último, Moodle requiere un espacio de almacenamiento, conocido como directorio de datos, donde almacena físicamente los ficheros utilizados en la plataforma: imágenes, manuales subidos por profesores, tareas realizadas por alumnos, etc.

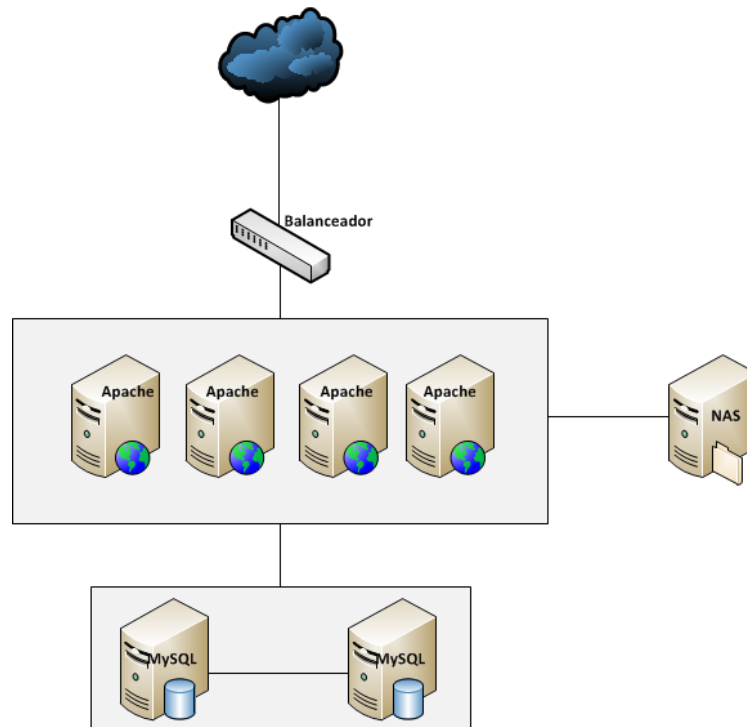


Figura 4-2 Infraestructura de las plataformas del Campus Virtual

4.4.3 Arquitectura de sistemas del Campus Virtual

Paralelamente e interoperando con el Campus Virtual hay otros sistemas en la ULPGC. La siguiente figura muestra los más importantes.

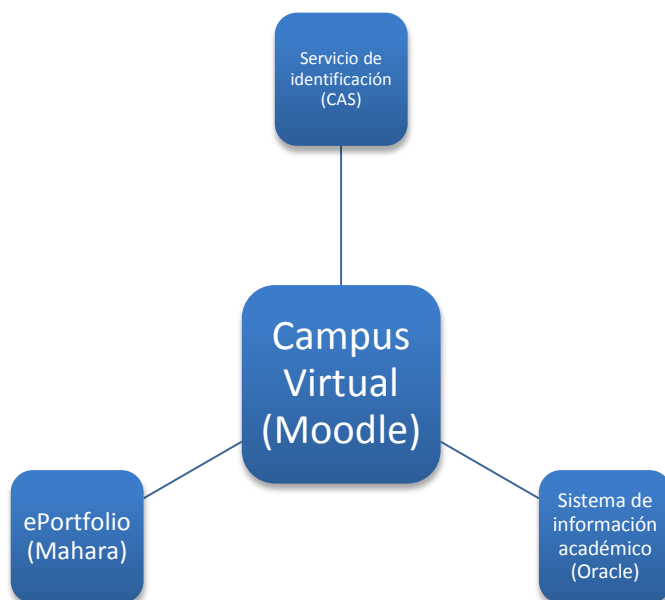


Figura 4-3 Arquitectura de sistemas de la ULPGC

- El servicio de identificación centralizado controla el acceso al resto de sistemas.
- El sistema de información académico comprende las aplicaciones y la base de datos donde se registran las titulaciones que se ofertan, las asignaturas que se imparten, las matrículas de los alumnos, etc.
- Mahara es un ePortfolio, donde los estudiantes y profesores pueden reunir sus logros y trabajos realizados, haciéndolos públicos a quien consideren oportuno, pudiendo elaborar su curriculum vitae o mostrar evidencias de conocimientos adquiridos.

5 HERRAMIENTAS Y TECNOLOGÍAS

5.1 SUBVERSION

La ULPGC utiliza Subversion como sistema de control de versiones. El código fuente de Moodle se gestiona en Subversion, por lo que su uso es condición indispensable para realizar este proyecto, puesto que las modificaciones se han de realizar siempre sobre la última versión del código de Moodle que esté usando la ULPGC y, al finalizar el proyecto, las extensiones desarrolladas deben estar integradas en Subversion.

5.1.1 Repositorio

El repositorio es el núcleo de Subversion, el almacén central de información en forma de árbol jerárquico de directorios y archivos. Los clientes se conectan al repositorio para obtener las distintas versiones del código e información sobre las mismas, así como para hacer públicos los cambios que han realizado.

Siguiendo la recomendación de la comunidad Subversion, que se puede considerar un estándar, cada proyecto cuenta con un directorio, y, dentro del mismo, tres subdirectorios:

- trunk: línea de desarrollo base, donde tiene lugar el desarrollo principal del proyecto
- tags: versiones del proyecto finales, se crean o se destruyen, pero no se modifican
- branches: líneas de desarrollo alternativas.

Para el desarrollo del proyecto se utilizará la rama ***branches/moodlepfc***, dentro del repositorio destinado al código del Campus Virtual.

En el Anexo II se define el protocolo de trabajo a seguir con Subversion.

5.2 ECLIPSE

El uso de un entorno de desarrollo integrado, IDE por sus siglas en inglés (*Integrated Developmet Environment*), aumenta considerablemente la productividad de un desarrollador, principalmente facilitando la edición de código, gestionando el acceso al servidor y mejorando la organización del código.

Eclipse es, probablemente, el IDE de código abierto más potente y con más usuarios del mercado. Cuenta con numerosas características y es altamente configurable. Hay disponibles una gran cantidad de plugins que amplían su funcionalidad, uno de los cuales, PHP Development Tools (PDT), convierte Eclipse en un formidable IDE para desarrollo PHP.

El Anexo III detalla la configuración de Eclipse utilizada para desarrollar este proyecto.

5.3 ENTORNO DE DESARROLLO

Para el desarrollo y prueba de las nuevas extensiones de Moodle se utiliza un entorno de desarrollo similar a las instalaciones de Moodle que se encuentran en producción: un entorno LAMP (Linux + Apache + MySQL + PHP) con una instalación de Moodle.

La infraestructura física la facilita para el desarrollo de este proyecto el Servicio de Informática (SI) de la ULPGC, habilitando una máquina al efecto y un usuario con permisos de administración con el que podemos instalar una versión de Moodle similar a las que se ofrecen a los usuarios del Campus Virtual de la ULPGC.

5.3.1 PHP

PHP es un lenguaje de código abierto del lado del servidor muy potente que se puede utilizar independientemente desde línea de comandos o integrado en un servidor web, permitiendo crear sitios con contenido dinámico. Es un lenguaje interpretado, una ventaja para los programadores, pues evita la necesidad de compilar antes de ejecutar el código; la edición y ejecución del código es mucho más rápida.

Para ejecutar Moodle 2.6, versión de Moodle instalada en el Campus Virtual de la ULPGC durante la realización del proyecto, se necesita como mínimo la versión de PHP 5.3.3.

La instalación y configuración de PHP corre a cargo del Área de Sistemas del Servicio de Informática de la ULPGC. Como referencia, consultar la documentación oficial de Moodle en el siguiente enlace https://docs.moodle.org/26/en/PHP_settings_by_Moodle_version.

5.3.2 MySQL y SQL

MySQL es un sistema de gestión de base de datos relacional (RDBMS). Básicamente, MySQL permite a los usuarios almacenar información en una estructura en forma de tabla, usando filas y columnas para organizar diferentes datos. Hay otros muchos RDBMS, pero esta es la escogida por la ULPGC tanto para sus aplicaciones corporativas como para el Campus Virtual.

PHP dispone de librerías para interactuar con MySQL, siendo necesario un conocimiento básico de SQL (*Structured Query Language*) para acceder y/o modificar los datos en MySQL. SQL es un pequeño lenguaje de fácil aprendizaje y uso, que permite:

- crear bases de datos
- crear tablas en una base de datos
- insertar datos en las tablas
- obtener datos de las tablas
- actualizar datos en las tablas

6 DESARROLLO EN MOODLE

Moodle ofrece una serie de directrices y guías para el desarrollo de código que se pueden encontrar en la documentación oficial (Moodle, 2014), completando esa información para redactar este capítulo con (Hunt, 2012).

En este apartado solo se incluye una pequeña introducción para facilitar la comprensión de las particularidades del desarrollo en Moodle, añadiéndose al final de esta memoria un anexo profundizando en cada aspecto relevante.

6.1 ARQUITECTURA DE MOODLE

6.1.1 Entorno de ejecución

El entorno escogido por la ULPGC para dar soporte a su Campus Virtual, siendo a su vez el más extendido, es el conocido como LAMP, *framework* de código abierto compuesto por Linux (sistema operativo), Apache (servidor web), MySQL (base de datos) y PHP (lenguaje de programación). Este último es el único indispensable, puesto que hay varias alternativas al sistema operativo, el servidor web y la base de datos sobre las que es operativo Moodle.

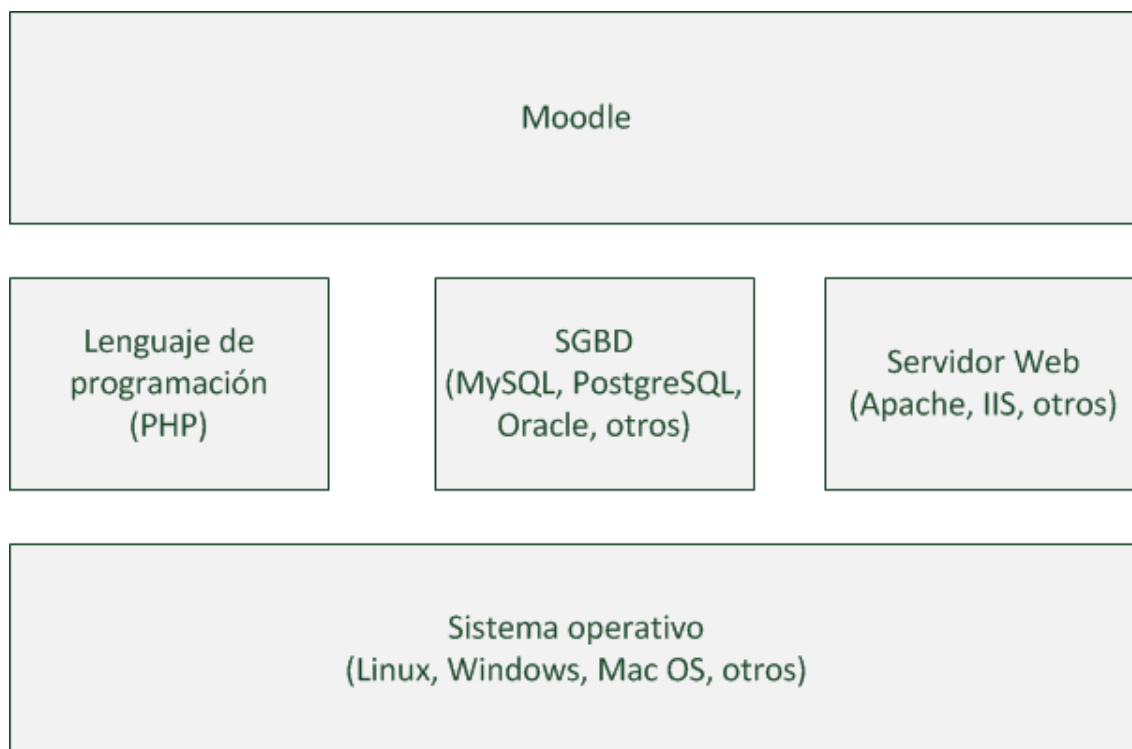


Figura 6-1 Entorno de ejecución

6.1.2 Capas de Moodle

Moodle se organiza en dos capas para ofrecer su funcionalidad: el código fuente (PHP, HTML, CSS y JavaScript) y los datos, repartidos en la base de datos y los ficheros de datos.

Estos componentes pueden estar en un solo servidor, pero en grandes instalaciones, se opta por un entorno con redundancia y balanceo de carga, con varios servidores web con una copia del código y una o varias instancias tanto de la base de datos como de los ficheros de datos.

La configuración global de Moodle, donde se especifica donde se encuentran los elementos mencionados, se almacena en un fichero llamado *config.php*, en el directorio raíz del código fuente.

Código fuente

El código fuente lo componen las librerías de uso general, los módulos, bloques, plugins y otras entidades. Se almacena en el sistema de ficheros en un directorio conocido como *dirroot*, que se especifica durante la instalación de Moodle. A continuación se muestra la organización de este directorio:

Carpeta	Funcionalidad
admin	Administración de Moodle
auth	Extensiones de autenticación
backup	Operaciones de copia y restauración
blocks	Bloques que se pueden colocar en los cursos y en la página principal
blog	Funcionalidad del blog
calendar	Gestión del calendario y de eventos
cohort	Manejo de grupos a nivel de sitio
comment	Comentarios utilizados en los cursos
course	Gestión de categorías, cursos y formatos de curso
enrol	Extensiones de matriculación
error	Manejo de errores
files	Gestión de ficheros
filter	Filtros aplicados a texto creado en el editor integrado
grade	Gestión de las calificaciones y del libro de calificaciones
group	Gestión de grupos y agrupamientos
install	Instalación y actualización de Moodle
iplookup	Búsqueda de direcciones IP
lang	Cadenas de idioma; hay una carpeta por cada uno
lib	Librerías del núcleo de Moodle
local	Directorio destinado a modificaciones locales
login	Gestión de acceso y creación de cuentas
message	Herramienta de mensajería que soporta varios canales
mnet	Conexión con otras instalaciones de Moodle
mod	Módulos incluidos por defecto en Moodle
my	Escritorio personal del usuario, conocido como myMoodle
notes	Gestión de notas en los perfiles de usuario
pix	Gráficos genéricos del sitio
plagiarism	Extensiones de detección de plagio
portfolio	Extensiones de portfolio que permiten a los usuarios exportar datos
question	Gestión del banco de preguntas, preguntas y categorías de pregunta
rating	Puntuaciones utilizadas en foros, glosarios y bases de datos
repository	Extensiones de repositorio: permiten importar y exportar datos
rss	Feeds RSS
search	Búsqueda local de cursos y global del sitio
sso	Operaciones Single sign-on
tag	Etiquetado
theme	Temas para cambiar la apariencia del sitio
user	Gestión de usuarios
webservice	Funcionalidad de Servicios Web

Tabla 3 Organización del código fuente

Datos

En la base de datos se guarda toda la información relacionada con los cursos, usuarios, roles, calificaciones, recursos añadidos por los profesores y la configuración del sistema. Por otro lado,

los ficheros tales como imágenes, documentos de texto u hojas de cálculo se almacenan en otro directorio de Moodle, conocido como *dataroot*.

Moodle gestiona internamente los ficheros, almacenando su ubicación física e información adicional (como el nombre, licencia, fecha de última modificación, etc.) en una tabla de la base de datos. Manipular ficheros directamente en el sistema de ficheros puede provocar comportamientos inesperados e incluso el mal funcionamiento de Moodle.

El directorio de datos, *dataroot*, donde se almacenan físicamente los archivos subidos a Moodle se organiza en las siguientes carpetas:

Carpeta	Archivos almacenados
filedir	Contenido de los usuarios, ficheros subidos por los mismos.
repository	Ubicación externa accesible desde Moodle
search	Ficheros temporales creados durante la realización de búsquedas
temp	Archivos temporales
trashdir	Ficheros eliminados

Tabla 4 Organización del directorio de datos

6.1.3 Moodle como sistema modular

Moodle cuenta con un núcleo de aplicación y numerosas extensiones (plugins) que ofrecen funcionalidad adicional. Está diseñado para ser personalizable y añadir capacidades sin modificar las librerías del núcleo, evitando problemas cuando se actualiza la aplicación a una nueva versión. De esta forma, cuando se quiere personalizar o extender una instalación de Moodle, siempre debe hacerse utilizando la arquitectura de extensiones.

Este modelo, común a muchos proyectos de código abierto, permite a los usuarios personalizar el sistema ajustándolo a sus necesidades. Moodle cuenta con un núcleo pesado, incluye mucha funcionalidad, y extensiones fuertemente tipadas, dependiendo de la funcionalidad a implementar, es necesario escribir un tipo determinado de extensión. Por ejemplo, un módulo de Actividad será muy diferente de una nueva extensión de Autenticación o un nuevo tipo de pregunta.

La tendencia en las últimas versiones de Moodle apuesta por reducir el núcleo moviendo funcionalidad a las extensiones. También se pretende estandarizar los diferentes tipos de extensiones tanto como sea posible.

Como ya se ha mencionado, hay diferentes tipos de extensiones. Una extensión de autenticación y otra de actividad interactúan con el núcleo de Moodle utilizando diferentes APIs, de acuerdo a la funcionalidad que la extensión añade. Sin embargo, los aspectos comunes de todas las extensiones (instalación, actualización, permisos, configuración, etc.) se manejan de igual forma con independencia del tipo de extensión.

Una distribución estándar de Moodle incluye las librerías del núcleo y un número de extensiones de cada tipo, siendo utilizable desde el momento que se instala. Tras la instalación, un sitio de Moodle se puede adaptar para un propósito específico cambiando las opciones de configuración por defecto y añadiendo o eliminando extensiones. La mayoría de las extensiones de dominio público están listadas en el directorio de extensiones de Moodle (<https://moodle.org/plugins/>).

Físicamente, una extensión de Moodle es básicamente una carpeta con scripts PHP, adicionalmente ficheros CSS y Javascript si es necesario. La comunicación del núcleo con la extensión se realiza utilizando puntos de entrada definidos, normalmente, en el archivo *lib.php*, perteneciente a la extensión.

Los tipos de extensión más importantes en Moodle son: actividades y recursos, bloques, temas, paquetes de idioma, formatos de curso, extensiones de autenticación, matriculación y de repositorios. Para obtener una lista actualizada de los tipos de extensiones disponibles, está disponible la función *get_plugin_types()*, definida en *lib/moodlelib.php*.

Cada extensión tiene un tipo y un nombre, cuya combinación generan el nombre de componente de la extensión en *Frankenstyle*, propio de Moodle. Asimismo, determinan la ruta de la extensión en el código fuente:

Tipo de extensión	Nombre extensión	Frankenstyle	Ruta
mod (módulo Actividad)	forum	mod_forum	mod/forum
mod (módulo Actividad)	quiz	mod_quiz	mod/quiz
block (bloque lateral)	navigation	block_navigation	blocks/navigation
qtype (tipo Pregunta)	shortanswer	qtype_shortanswer	question/type/shortanswer
quiz (informe Cuestionario)	statistics	quiz_statistics	mod/quiz/report/statistics

Tabla 5 Ejemplos de extensiones

El último ejemplo muestra cómo para un módulo de actividad se pueden declarar extensiones de nivel inferior. Por el propio diseño de Moodle, el uso de extensiones dentro de extensiones genera problemas de rendimiento, por lo que lo es posible realizarlo para los módulos de actividad, dado que son los que permiten desarrollar actividades educativas, eje de un LMS.

6.1.4 Núcleo de Moodle

Provee la infraestructura necesaria para construir un Learning Management System. Implementa los conceptos clave que las diferentes extensiones necesitan para interactuar con el sistema. Ver el apartado 3.1 para obtener una descripción detallada de estos conceptos.

Cursos y usuarios

- **Cursos y actividades:** a curso de Moodle es un conjunto de actividades y recursos agrupados en secciones. Los cursos se organizan jerárquicamente en categorías.
- **Usuarios:** cualquiera que usa el sistema. Hay varios roles definidos por defecto en los que se puede clasificar un usuario: Estudiantes, Profesores y Administradores. Se pueden crear nuevos tipos de usuario.
- **Capacidades:** una capacidad describe alguna funcionalidad particular de Moodle. Las capacidades se asocian a roles.
- **Contexto:** es un espacio en Moodle, como un curso, una actividad, un bloque, etc.
- **Permisos:** es un valor asignado a una capacidad para un rol en concreto.

- **Roles:** un rol identifica el estatus del usuario en diferentes contextos, las capacidades que tiene en el mismo.

Funcionalidad incluida

- **Creación y edición de perfiles de usuario:** en el momento que se crea una cuenta de usuario, se crea un perfil para el mismo. El usuario debe incluir algunos detalles para completar su perfil. Normalmente, un usuario tiene permiso para editar su propio perfil en cualquier momento.
- **Grupos y cohortes:** las cohortes son grupo a nivel de sitio. Permite tratar a todos los miembros de una cohorte en un curso realizando una sola acción.
- **Matriculación y control de acceso:** los usuarios se matriculan en diferentes cursos y, acorde al rol asignado y a los grupos a los que pertenece, tienen permiso para realizar determinadas acciones.
- **Finalización de actividades y cursos:** el sistema de finalización de actividad permite marcar como finalizadas actividades solo cuando se cumplen determinadas condiciones.
- **Paneles de navegación y administración:** el bloque de navegación facilita el acceso a varias secciones de Moodle, como la página inicial, el perfil del usuario, las páginas del sitio y los cursos. Por otro lado, desde el panel de administración se pueden configurar todas las opciones de Moodle.
- **Librería Javascript:** Moodle utiliza Yahoo User Interface (YUI library). Dispone asimismo de un mecanismo para cargar los archivos Javascript adicionales que sean necesarios en cada página.
- **Logs y estadísticas:** los logs son informes de la actividad de los usuarios. Están disponibles a nivel de sitio y de curso. Las tablas y gráficas de estadísticas muestran el uso de diferentes componentes del sitio en periodos de tiempo concretos.

6.2 ORGANIZACIÓN DEL CÓDIGO

Moodle sigue una aproximación **transaction script**, para más información <http://martinfowler.com/eaCatalog/transactionScript.html>. Supongamos que se visualiza un foro. La URL sería `.../mod/forum/view.php?id=1234`, y `mod/forum/view.php` el script PHP que genera la página.

Detrás de esta simple aproximación, gran parte de la funcionalidad básica de Moodle se ha agrupado en librerías, en su mayoría en la carpeta *lib*. Provee ciertos elementos de un modelo de dominio (<http://martinfowler.com/eaCatalog/domainModel.html>). Moodle, como proyecto, se inició antes de que PHP fuese orientado a objetos, por lo que dispone de modelos de dominio orientados a objetos excepto en el código desarrollado recientemente.

Hay dos capas que separan la presentación de la lógica del negocio. La capa superior es el tema, que controla los aspectos más visuales de la interfaz de Moodle. La otra capa la forman las clases **render**, que generan el código HTML a partir de los datos procedentes de los transaction scripts y el modelo del dominio. Desafortunadamente, ni PHP ni la arquitectura de Moodle establecen una clara separación de la capa UI (*User Interface*).

6.3 BASE DE DATOS

Comprende más de 250 tablas, ya que la base de datos en su conjunto es la unión de las tablas del núcleo y las tablas de cada extensión. Esta enorme estructura es comprensible puesto que las tablas de una extensión particular, por lo general, solo se relacionan entre sí y con alguna tabla del núcleo.

Moodle tiene una potente capa de abstracción de base de datos desarrollada a medida, llamada **XMLDB**, que permite que el mismo código se ejecute en diferentes bases de datos. Se han definido distintas herramientas así como una API para definir y modificar tablas y añadir y obtener datos de la base de datos.

La estructura de la base de datos está definida en los ficheros *install.xml* en la carpeta *db* de cada extensión. Por ejemplo, *mod/forum/db/install.xml* contiene la definición de la base de datos para el módulo foro. *lib/db/install.xml* define las tablas usadas por el núcleo de Moodle. El fichero *install.xml* incluye comentarios que explica el propósito de cada tabla y columna. Estos comentarios se pueden convertir en documentación legible en **Administración del sitio – Desarrollo - Editor XMLDB**, pulsando el enlace **[Doc]**.

6.3.1 La clase *moodle_database*

La clase *moodle_database* define la interfaz de acceso a base de datos, y se accede a ella usando la variable global *\$DB*:

```
$course = $DB->get_record('course', array('id' => $courseid));
```

Esta llamada genera la sentencia SQL:

```
SELECT * FROM mdl_course WHERE id = $courseid;
```

Devolviendo los datos como un objeto PHP plano con campos públicos, permitiendo acceder con `$course->id`, `$course->fullname`, etc.

Los métodos más simples como el anterior son útiles para realizar consultas sencillas. En el caso de consultas más complejas, hay métodos para ejecutar cualquier sentencia SQL. Para conocer todos los métodos disponibles, consultar la documentación de Moodle (Moodle, 2014).

Tener en cuenta:

- Los nombres de las tablas se encierran en {}, de tal forma que la librería las identifique y añada el prefijo que corresponda. El prefijo de las tablas de la base de datos de Moodle se puede definir durante el proceso de instalación.
- La librería utiliza parámetros de sustitución (*placeholders*) para incluir valores en la sentencia SQL. Acepta tanto parámetros con nombre como parámetros anónimos, usando ? como parámetro.
- Para que las consultas sea soportadas por todas las bases de datos, es necesario utilizar un subconjunto de SQL.
- Si no existiese un estándar de SQL para realizar una sentencia, hay funciones compatibles que generan el SQL correcto. Por ejemplo, la concatenación de cadenas difiere según SGBDR.

6.4 ESTILO DEL CÓDIGO

Un estilo de código consistente es importante en cualquier proyecto de desarrollo de software, sobre todo cuando hay muchos desarrolladores implicados. Seguir un estilo estándar facilita que el código sea más sencillo de leer y comprender, mejorando la calidad global del software.

En el Anexo IV se incluye la Guía de estilo de código de Moodle, a la cual se ciñe el desarrollo de este proyecto.

6.5 SEGURIDAD

Es necesario proteger los intereses e información de todos los usuarios. Una instalación de Moodle puede contener información de carácter sensible, como discusiones privadas y puntuaciones. Adicionalmente, es necesario proteger a los usuarios de spammers y otros usuarios indeseados de internet.

Moodle no deja de ser un script ejecutándose en un servidor, por lo que no debe introducir vulnerabilidades que podrían permitir a hackers obtener acceso al servidor en el que se ejecuta. Cualquier script, ya sea en el núcleo o en una aportación de terceros, debe seguir estrictamente la guía de seguridad de Moodle, un extracto de la cual se está disponible en el Anexo V.

6.6 INTERNACIONALIZACIÓN

Moodle se utiliza en más de 84 idiomas, por lo que es necesario prestar especial atención en mantener las cadenas de idioma y la información local separada del código, en paquetes de idiomas. El idioma por defecto para todo el código, comentarios y documentación es el inglés.

Moodle utiliza su propio sistema para traducir la interfaz en cualquier idioma. El sistema se base en el uso de la función **get_string**. Las cadenas se identifican con una clave y el nombre *Frankenstyle* de la extensión. Es posible intercalar texto dentro de las cadenas.

Las cadenas en los diferentes idiomas se almacenan en ficheros que contienen arrays PHP planos. Los diferentes idiomas se identifican por el código de dos letras del país. Un paquete de idioma puede derivarse de otro, de tal forma que solo es necesario definir aquellas cadenas que difieran del paquete de idioma original. Por ejemplo, el paquete *fr_ca* (francés de Canadá) declara *fr* (francés) como el idioma padre.

La API de **get_string** oculta complejidad al desarrollador, como determinar el idioma actual (en base a las preferencias del usuario, la configuración del curso actual o la configuración global) y la búsqueda en los diferentes paquetes de idioma para encontrar la cadena a traducir.

6.7 RENDIMIENTO

La carga que una instalación de Moodle puede soportar depende, principalmente, de los servidores y el hardware de red sobre el que se ejecuta. Sin embargo, unas buenas prácticas aplicadas al desarrollo de código pueden mejorar sensiblemente su desempeño.

Moodle ofrece una serie de recomendaciones orientadas a la generación de páginas solicitadas por los usuarios. En el caso de los scripts a ejecutar desde consola, habrá que tener en cuenta:

- Limitar la cantidad de RAM que consumen, para no saturar el servidor web y finalizar anormalmente la ejecución del script.
- Realizar el menor número de consultas necesario a la base de datos.
- Minimizar el impacto en el uso cotidiano por parte de los usuarios.

7 ANÁLISIS

Este apartado contiene la especificación de requisitos y toda la documentación del análisis, a partir de la cual se elaborará posteriormente el diseño.

7.1 DEFINICIÓN DEL SISTEMA

El sistema al completo comprende la creación y carga del Data Mart intermedio, con los datos a sincronizar ya procesados, y los scripts de Moodle que realizan la sincronización.

Aunque he desarrollado ambos, cada uno de ellos tiene entidad para ser un proyecto por sí mismo y, de hecho, se realizaron como proyectos independientes que luego se integraron.

7.1.1 Determinación del Alcance del Sistema

El propósito de este proyecto es el desarrollo de las extensiones necesarias en Moodle para que, a partir de la información contenida en el sistema de información de la ULPGC, el Campus Virtual de la ULPGC sea un reflejo de la realidad académica.

En un sistema de información tan complejo son muchas las particularidades a tener en cuenta, escapando muchas de ellas al propósito general de este proyecto, relegándose a desarrollos posteriores.

El énfasis de este proyecto recae en el análisis de la información necesaria para cargar correctamente el Campus Virtual y el desarrollo de las extensiones acorde a las directrices de desarrollo de Moodle, de tal forma que su utilidad no se reduzca al ámbito de la ULPGC, sino que puedan ser reutilizados en condiciones similares en otros ámbitos.

El proceso de sincronización entre la información de la ULPGC y el Campus Virtual será completamente automático y desatendido, por lo que no se contemplan interfaces de usuario ni la interacción bilateral entre ambos sistemas. Asimismo, nos centraremos en solo una de las plataformas del Campus Virtual, considerando que la aplicación al resto de ellas es similar. Escogemos la plataforma de Grado y Posgrado por ser la más compleja, ya que comprende el mayor número de cursos y usuarios.

La información que se sincronizará abarca centros, titulaciones, asignaturas, grupos, profesores, alumnos y matrículas.

7.2 REQUISITOS DEL SISTEMA

7.2.1 Centros

Los centros se encargan de la organización de las enseñanzas y de los procesos académicos, administrativos y de gestión conducentes a la obtención de los títulos. Pueden ser Facultades o Escuelas, estando dirigidas por un decano o director. El primer nivel de organización dentro del Campus Virtual serán los centros. Debe existir una categoría por cada Facultad, Escuela o Instituto que imparta alguna titulación.

Identificador	F01	Nombre	CrearCentro
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Crear una categoría representando un centro en el Campus Virtual		

Identificador	F02	Nombre	EliminarCentro
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Eliminar una categoría que represente a un centro del Campus Virtual, eliminando las titulaciones correspondientes		

7.2.2 Titulaciones

Las titulaciones, articuladas en planes de estudio, son el conjunto de enseñanzas organizadas por una Universidad cuya superación da derecho a la obtención de un título. Son los centros los que ofertan las distintas titulaciones, por lo que cada una de ellas estará asociada a un centro.

Las titulaciones conformarán el segundo nivel en la categorización de los cursos en el Campus Virtual. Deben estar asociadas al centro que las oferta. Cada curso estará en la titulación en la que se imparte la asignatura a la que representa. Existirá una categoría por cada titulación, dentro de la categoría correspondiente al centro que oferta la titulación.

Identificador	F03	Nombre	CrearTitulación
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Crear una categoría que represente una titulación en el Campus Virtual, dentro de la categoría del centro que la oferta		

Identificador	F04	Nombre	EliminarTitulación
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Eliminar la categoría que representa la titulación del Campus Virtual, eliminando las asignaturas asociadas		

7.2.3 Asignaturas

Corresponde a los centros proponer las asignaturas que componen un plan de estudio, estando orientadas al cumplimiento de los objetivos del mismo. Hay que tener presente la existencia de asignaturas vinculadas. Son aquellas que están asociadas a otra asignatura, definida como maestra, que es la que realmente se imparte. Las asignaturas se asignan a los departamentos que se consideren competentes para impartirla.

Todas las asignaturas que se imparten en la UPLGC deben tener presencia en el Campus Virtual, en la forma de cursos. Es importante conocer la relación con el departamento responsable. Para las asignaturas vinculadas no se creará un curso.

Identificador	F05	Nombre	CrearAsignatura
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Añadir un curso que represente una asignatura en el Campus Virtual, asociada la titulación en la que se imparte		

Identificador	F06	Nombre	EliminarAsignatura
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Elimina un curso del Campus Virtual		

7.2.4 Grupos

Los alumnos dentro de cada asignatura se organizan en grupos. Existen dos tipos de grupos:

- docencia: útiles para estructurar la impartición de las clases. Divide a los alumnos en varios grupos de teoría y prácticas. Cada grupo de docencia tiene un profesor asignado, encargado de impartir la docencia en el mismo;
- actas: orientados a la calificación final de los alumnos por parte de los profesores. No es necesario que tengan un profesor asignado.

En cada curso dentro del Campus Virtual deben estar definidos los grupos de las asignaturas.

Identificador	F07	Nombre	CrearGrupo
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Crea un grupo dentro del curso que representa la asignatura a la que pertenece el grupo, identificándolo como grupo académico		

Identificador	F08	Nombre	EliminarGrupo
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Elimina un grupo del curso que representa la asignatura a la que pertenece		

A cada grupo hay que asignar tanto a los profesores como a los alumnos que correspondan.

Identificador	F09	Nombre	AsignarGrupo
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Asignar los usuarios al grupo		

Identificador	F10	Nombre	DesasignarGrupo
Tipo	Funcional	Fecha	20/03/2013
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Desasignar los usuarios que dejen de pertenecer a un grupo		

7.2.5 Usuarios

Debe tener acceso al Campus Virtual todo docente y alumno de la ULPGC, autenticándose previamente en el Servicio de Identificación Centralizada. No pueden identificarse ni crear una cuenta directamente en el Campus Virtual.

Por el contrario, un alumno que anula matrícula o un profesor que deja de impartir docencia pierde el derecho a acceder al Campus Virtual. No pueden anular su cuenta motu proprio.

La sincronización de cuentas se realizará diariamente, de tal forma que un usuario pueda acceder al campus al día siguiente de haber formalizado la matrícula. El proceso se realizará en el horario de menor uso, para reducir el impacto en el rendimiento de la plataforma.

Identificador	F11	Nombre	AltaUsuario
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Creación de cuenta de usuario en el Campus Virtual, con sus datos personales actualizados. Los usuarios no pueden crear cuentas por sí mismos. Si el usuario fue creado manualmente, identificarlo como usuario sincronizado.		

Identificador	F12	Nombre	BajaUsuario
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Eliminar cuenta de usuario en el Campus Virtual. Los usuarios no pueden eliminar su cuenta por sí mismos.		

Identificador	F13	Nombre	IdentificaciónUsuario
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Los usuarios se debe identificar en el Servicio de Identificación Centralizada de la ULPGC, no directamente en el Campus Virtual		

7.2.6 Matrículas

Cada profesor debe poder acceder y modificar aquellos cursos en el Campus Virtual correspondiente a las asignaturas que imparte en la ULPGC. Asimismo, cada alumno debe poder acceder y participar en aquellos cursos que representan a las asignaturas en las que está matriculado. Igualmente, si un alumno causa baja en una asignatura o un profesor deja de tener docencia en una asignatura, se desmatricula del curso correspondiente en el Campus Virtual.

Identificador	F14	Nombre	MatricularUsuario
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Se matricula el usuario en el curso en el que participa con el rol correspondiente. Si se hubiese matriculado previamente de forma manual, identificar la matrícula como procedente de la sincronización.		

Identificador	F15	Nombre	DesmatricularUsuario
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Se desmatricula un usuario de un curso		

7.2.7 Global

La sincronización de la información se realizará diariamente, de tal forma que los cambios en el sistema de información de la ULPGC se reflejen, como máximo, al día siguiente. El proceso de sincronización se realizará en el horario de menor uso, para reducir el impacto en el rendimiento del Campus Virtual.

Identificador	F16	Nombre	SincronizaciónDiaria
Tipo	Funcional	Fecha	20/03/2014
Prioridad	Alta	Necesidad	Sí
Estabilidad	Normal	Verificable	Sí
Descripción	Sincronización diaria, en el horario que menos afecte el rendimiento del Campus Virtual		

7.3 IDENTIFICACIÓN Y DESCRIPCIÓN DE LOS SUBSISTEMAS

Hay dos subsistemas bien diferenciados:

- El sistema de información de la ULPGC, en adelante SI de la ULPGC, que proporciona la información académica de la universidad; su gestión queda fuera del ámbito de este proyecto.
- Moodle, donde se deben implementar los mecanismos necesarios para cargar la información facilitada por el SI de la ULPGC.

7.3.1 Sistema de Información de la ULPGC

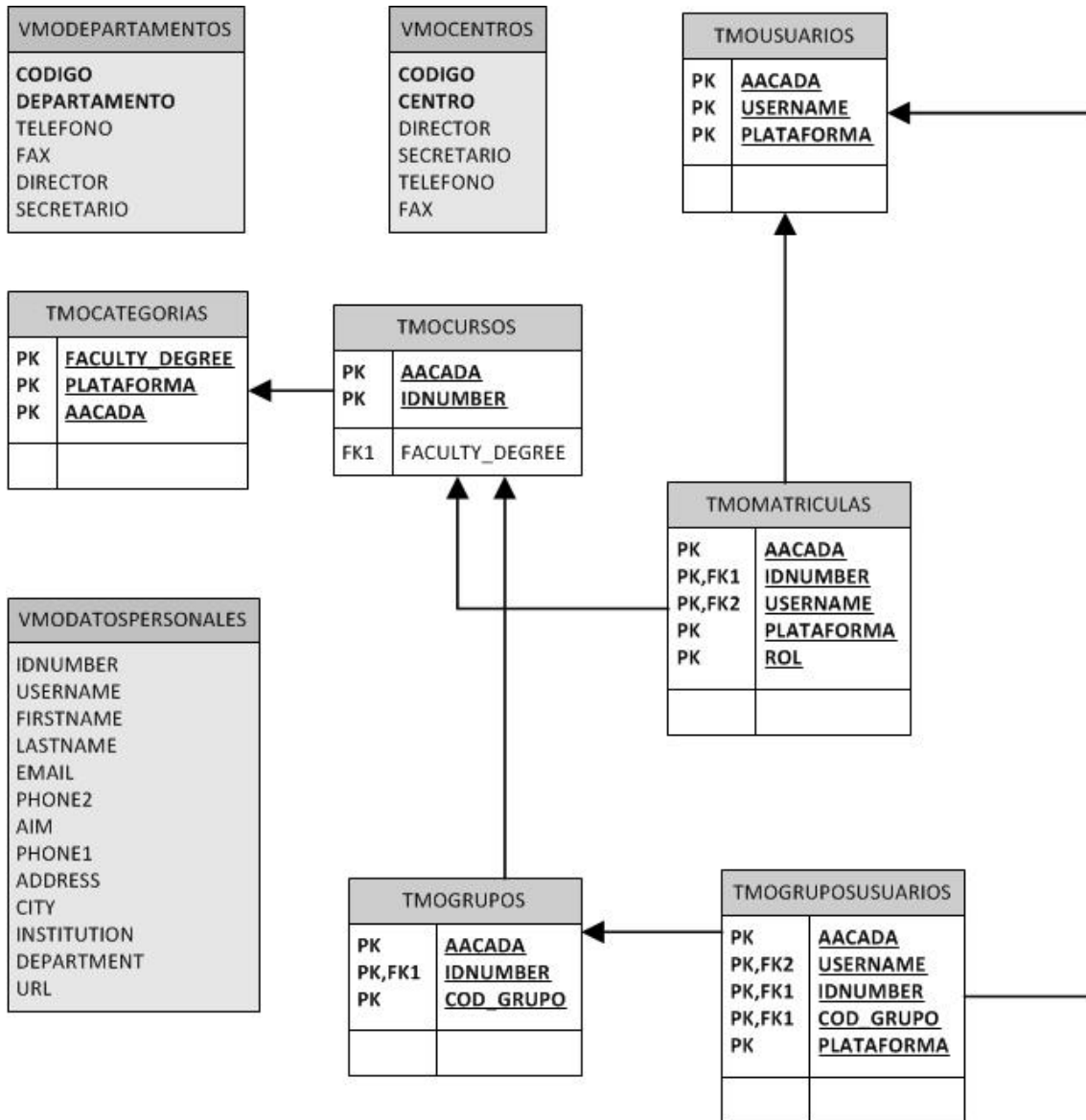
Como se mencionó en el apartado 4.3, el SGBD Oracle es la base del sistema de información de la ULPGC. De él habrá que extraer toda la información necesaria para montar la estructura del Campus Virtual.

La información académica que requiere el Campus Virtual no se puede obtener directamente de la base de datos institucional, sino que requiere su extracción y transformación antes de poder procesarla. Para realizar esta tarea, será necesario crear un sistema intermedio (*Data Mart*), donde se almacenará la información procesada para ser directamente integrada en el Campus Virtual. En base a este criterio, diferenciaremos:

- Esquemas, tablas y vistas de las que se extrae la información académica que debe reflejarse en el Campus Virtual (cursos, profesores, alumnos matriculados, etc.);
- Esquema intermedio con las tablas, vistas y procedimientos utilizados para adecuar la información al formato que requiere el Campus Virtual.

Esquema de la base de datos

El esquema correspondiente contiene múltiples paquetes PL/SQL (un paquete PL/SQL agrupa tipos de datos, procedimientos y funciones), vistas y tablas. En la siguiente figura se muestran las tablas y vistas desde las que directamente se obtendrá la información a incluir en el Campus Virtual. No se incluyen todos los campos puesto que no son relevantes, solo aquellos que forman parte de las claves primarias y secundarias.



Los nombres de las tablas y vistas son bastante descriptivos:

- **VMODEPARTAMENTOS:** un registro por cada departamento de la ULPGC.
- **VMOCENTROS:** un registro por cada centro de la ULPGC.

- **VMODATOSPERSOANALES:** para los datos personales se utiliza una vista para tener siempre los datos más actualizados de los usuarios.
- **TMO CATEGORIAS:** un registro por cada entidad de la ULPGC que debe tener una categoría en el Campus Virtual, titulaciones y centros en su mayoría.
- **TMO CURSOS:** un registro por cada entidad de la ULPGC que debe tener un curso en el Campus Virtual, asignaturas principalmente.
- **TMO USUARIOS:** un registro por cada usuario y plataforma en la que debe tener cuenta.
- **TMO MATRICULAS:** un registro por cada asignación de un usuario a un curso en el Campus Virtual, como es el caso de los profesores y alumnos que participan en cada asignatura.
- **TMO GRUPOS:** un registro por cada grupo que se debe crear en un curso en el Campus Virtual, por ejemplo, los grupos de docencia y de actas.
- **TMO GRUPOS USUARIOS:** un registro por cada asignación de un usuario a un grupo, ya sea profesor o estudiante.

7.3.2 Moodle

Según se describe en la documentación oficial (Moodle, s.f.), Moodle está estructurado en torno a un núcleo de la aplicación, complementado con varias extensiones (plugins), cada una de las cuales proporciona una funcionalidad específica. Moodle está diseñado para ser altamente ampliable y personalizable sin modificar las librerías del núcleo, lo que provocaría problemas en el proceso de actualización a nuevas versiones. De esta manera, las modificaciones o personalizaciones sobre una instalación de Moodle siempre se deben hacer utilizando extensiones.

Hay diferentes tipos de extensiones en Moodle. Una extensión de identificación (autenticación) y un módulo de actividad se comunican con el núcleo de Moodle utilizando diferentes APIs, acorde a la funcionalidad que la extensión provee. Las funcionalidades comunes a todas las extensiones (instalación, actualización, permisos, configuración,...) se manejan de forma similar en los diferentes tipos de extensiones.

Físicamente, una extensión de Moodle es sencillamente un directorio con scripts PHP (y archivos CSS y Javascript si son necesarios). El núcleo de Moodle se comunica con las

extensiones utilizando determinadas funciones, normalmente definidas en el archivo *lib.php* incluido en la extensión.

7.3.2.1 Extensiones de identificación

Es el mecanismo que permite a los usuarios acceder a un sitio Moodle en base a unas credenciales, normalmente un usuario y una contraseña. Además, gestiona la creación de cuentas de usuario, incluyendo la información personal del mismo, si no existe previamente. Se usarán indistintamente los términos *identificación* y *autenticación*.

Estado actual

La sincronización se realiza utilizando una extensión desarrollada a medida, que a partir de un fichero de texto crea y elimina las cuentas de usuario en el Campus Virtual. Este fichero de texto se genera en la base de datos corporativa e incluye todos los alumnos con matrícula vigente.

Comparando los usuarios en el fichero con las cuentas existentes en el Campus Virtual, se añaden aquellos que no están en el Campus Virtual y se eliminan los que no existen en el fichero.

Hay varios problemas asociados a este procedimiento:

- si el fichero se genera incorrectamente y no se incluyen todos los usuarios, éstos se eliminan del Campus Virtual;
- todos los días se traspasa la información de todos los alumnos, se hayan modificado o no, con el consiguiente impacto en el procesado del fichero;
- aunque el proceso está automatizado, implica la creación del fichero de texto en la base de datos corporativa, importarlo al servidor del Campus Virtual y luego procesarlo. En ocasiones este proceso hay que realizarlo manualmente, resultando engorroso;
- la extensión está desarrollada para la versión anterior de Moodle, por lo que es necesario adaptarla a la nueva versión.

Extensiones nativas

Moodle ofrece varias formas de gestionar la autenticación, utilizando distintas **extensiones de identificación** (plugins). Por defecto están disponibles las siguientes:

- *Cuentas manuales* - cuentas creadas manualmente por un administrador
- *No login* - cuenta de usuario suspendida

- *Identificación basada en email* - permite a los usuarios crear su cuenta
- *Usar servidor CAS (SSO)* - los detalles de las cuentas están en un servidor CAS externo
- *Usar una base de datos externa* - los detalles de las cuentas están en una base de datos externa
- *Usar servidor FirstClass* - los detalles de las cuentas están en un servidor externo FirstClass
- *Usar un servidor IMAP* - los detalles de las cuentas están en un servidor externo IMAP
- *Autenticación LDAP* - los detalles de las cuentas están en un servidor externo LDAP
- *Identificación Moodle Network* - how different Moodle sites can connect and authenticate users
- *Usar un servidor NNTP* - los detalles de las cuentas están en un servidor externo NNTP
- *No authentication* - utilizada para pruebas o si la instalación de Moodle no está online
- *PAM (Pluggable Authentication Modules)* - los detalles de las cuentas residen en el sistema operativo sobre el que está la instalación de Moodle, vía PAM (sólo Linux/Unix).
- *Servidor POP3* - los detalles de las cuentas están en un servidor POP3 externo
- *Servidor RADIUS* - los detalles de las cuentas están en un servidor RADIUS externo
- *Shibboleth* - los detalles de las cuentas están en un servidor Shibboleth externo

La instalación de Moodle puede tener activas todas las extensiones de identificación que se precisen, y cada cuenta de usuario debe estar asociada a una de ellas.

Además de las extensiones incluidas por defecto, se pueden añadir extensiones de terceros o desarrollarlas a medida.

Evaluación de las extensiones nativas

Cuentas manuales

La creación de cuentas corre a cargo de un administrador. La actualización de las mismas la pueden realizar tanto el administrador como el propietario de la cuenta.

Es una tarea inviable crear y mantener manualmente todas las cuentas de los alumnos y profesores de la ULPGC, ya que superan con creces los 20.000 usuarios.

Identificación basada en email

Esta extensión permite a los usuarios crear su propia cuenta, poniendo a disposición de los mismos la opción Crear nueva cuenta en la página de identificación. Tras rellenar los datos solicitados, recibirán un correo de confirmación en la cuenta de correo electrónico que hayan especificado.

El requisito F11 especifica que los usuarios no pueden crear cuentas por sí mismos en el Campus Virtual, por lo que esta extensión se descarta.

Métodos de identificación basados en servidores externos: FirstClass, IMAP, NTTP, POP3, Radius, Shibboleth

La ULPGC no dispone de estos servidores a efectos de autenticación, por lo que ni siquiera se valoran.

Usar servidor CAS (SSO)

CAS es muy útil en entornos donde los usuarios pueden acceder a diferentes aplicaciones web. Si todas las aplicaciones web estuviesen “CASificadas”, un usuario se identificaría solo una vez y podría acceder a todas las aplicaciones.

En el anexo 22 se documenta el funcionamiento de CAS. Básicamente basta con configurar una aplicación para no utilizar un mecanismo de autenticación propio, sino redirigir los usuarios sin identificar a un servidor CAS, que devolvería un token de autenticación a la aplicación original. Moodle puede extraer el nombre de usuario del token y utilizarlo en su mecanismo interno de autorización y atributos. La ventaja es que Moodle no tendría que manejar contraseñas y que los usuarios, una vez autenticados, pueden acceder a otras aplicaciones sin tener que volver a hacerlo.

Sin embargo, este mecanismo únicamente permite validar la identidad de un usuario, pero no crear las cuentas de usuario en Moodle.

Autenticación LDAP

El servidor LDAP de la ULPGC se utiliza únicamente a efectos de identificación, no conteniendo información relativa a las titulaciones en las que está matriculado un usuario (determina en qué plataforma debe existir), ni siquiera si está matriculado actualmente o no. Los requisitos F11 y F13 especifican que sólo los alumnos matriculados tendrán cuenta activa en el Campus Virtual, por lo que no se puede utilizar como fuente para la creación de cuentas de los mismos.

Por otro lado, el mecanismo de identificación centralizado de la ULPGC, se fundamenta en el servidor LDAP de la institución, por lo que indirectamente será necesario utilizarlo.

PAM

Este método permite acceder a los usuarios de Moodle utilizando las cuentas nativas de usuario en el servidor dónde se encuentra alojado Moodle. Esto implica la necesidad de crear y eliminar usuarios a nivel de sistema operativo, lo que añade un nivel más de sincronización: base de datos institucional, sistema operativo y Moodle deben tener las mismas cuentas de usuario.

Al igual que sucede con la creación de cuentas manuales, aunque en este caso se puede automatizar el proceso, la ingente cantidad de usuarios hace muy costoso el proceso: la creación y mantenimiento de más de 20.000 cuentas de usuario en el sistema operativo, teniendo además presente la existencia de varios frontales web.

Identificación Moodle Network (Mnet)

Permite establecer un vínculo con otras instalaciones de Moodle o Mahara, compartiendo recursos entre ellas y habilitando la autenticación de usuarios de una instalación en otra. Las plataformas del Campus Virtual son independientes en cuanto a contenido y usuarios, no dándose las condiciones para utilizar esta extensión.

Usar base de datos externa

Este método utiliza una base de datos externa para comprobar si un usuario tiene acceso Moodle. Si el usuario no existe en Moodle, se crea una nueva cuenta y su información se crea desde la base de datos externa en la base de datos de Moodle.

Este proceso únicamente crea cuentas cuando el usuario intenta identificarse. Si se desea que las cuentas se creen en Moodle independiente del proceso de identificación, esta extensión cuenta

con un script que se puede ejecutar regularmente para crear las nuevas cuentas, concordando con el requisito R05.

Además, permite mapear campos de la base de datos externa con valores del perfil del usuario, almacenado en la base de datos de Moodle. De esta forma, se puede identificar qué campos se desea mantener actualizados.

Esta extensión cumple con los requisitos R01, R02 y R03. Sin embargo, la autenticación se realiza directamente en Moodle, utilizando la contraseña que el usuario tiene en su perfil o, en caso de tratarse de un usuario nuevo, la contraseña almacenada en la base de datos externa, por lo que no satisface el requisito R04.

Conclusión

De la evaluación anterior, teniendo en cuenta que la información de los usuarios se almacena en la base de datos institucional pero que la autenticación pasa por utilizar el **Servicio de Identificación Centralizada**, podemos modificar ligeramente la extensión para usar una base de datos externa, cambiando el proceso de autenticación para utilizar el **Servicio de Identificación Centralizada** de la ULPGC, tal como implementa la extensión basada en CAS. A esta nueva extensión la denominaremos **CASulpgc**.

7.3.2.2 Extensiones de matriculación (enrolment)

Habilitar la participación de usuarios en cursos se conoce en Moodle como enrol o matriculación. Es, a todos los efectos, el equivalente a la matrícula de un alumno en una asignatura o la asignación de un profesor a un asignatura. En Moodle, la diferencia entre un profesor y un alumno radica en los permisos que tiene el usuario dentro del curso, representados por los roles. Los roles determinan que puede hacer un usuario dentro de un curso.

Las extensiones de matriculación controlan qué usuarios están matriculados en qué cursos y con qué roles. Como en el caso anterior, se puede sincronizar con otro sistema, como un sistema de información de estudiantes, o puede ser gestionado internamente por Moodle. Las extensiones de matriculación se encuentran en el directorio *enrol*.

Estado actual

Al igual que sucedía con la creación de cuentas de usuario, la matriculación de usuarios en cursos se realiza mediante un proceso de sincronización utilizando una extensión desarrollada a medida, que a partir de un fichero de texto asigna y desasigna roles a los usuarios dentro de un curso.

Este fichero de texto se genera en la base de datos corporativa e incluye la relación de matrículas de alumnos en asignaturas así como la asignación de profesores a las asignaturas.

Hay varios problemas asociados a este procedimiento:

- todos los días se traspasa la información relativa a matrículas de alumnos y asignación de profesores, lo que implica un procesado considerable;
- aunque el proceso está automatizado, implica la creación del fichero de texto en la base de datos corporativa, importarlo al servidor del Campus Virtual y luego procesarlo. En ocasiones este proceso hay que realizarlo manualmente, resultando engorroso;
- la extensión está desarrollada para Moodle 1.9.x, por lo que es necesario adaptarla a la nueva versión de Moodle.

Extensiones nativas

Moodle ofrece varias formas de gestionar la matriculación, utilizando distintas **extensiones de matriculación** (plugins). Por defecto están disponibles las siguientes:

- *Matriculación manual* - el administrador o gestor del curso añade a los usuarios manualmente
- *Auto-matriculación* - el usuario puede matricularse en un curso
- *Sincronización de cohorte* - los usuarios de una cohorte se matriculan en el curso (las cohortes son un concepto fuera del ámbito de este proyecto)
- *Metaenlace de curso* - matricula automáticamente participantes en un curso en otro
- *Acceso de invitados* - los usuarios pueden acceder al curso sin ser participantes
- *Categoría de matrícula* - los usuarios se matriculan en todos los cursos de una categoría
- *Base de datos externa* - los usuarios se matriculan según la información almacenada en una base de datos externa
- *Archivo plano (CSV)* - los usuarios se matriculan según la información almacenada en un fichero de texto plano
- *Archivo IMS Enterprise* - los usuarios se matriculan según la información almacenada en un fichero con este formato XML estándar

- *Matriculaciones LDAP* - los usuarios se matriculan según la información almacenada en un directorio LDAP
- *Matriculaciones remotas MNet* - los usuarios se matriculan en base a un sitio Moodle enlazado
- *Paypal* - los usuarios se matriculan tras pagar su inscripción con Paypal.

La instalación de Moodle puede tener activas todas las extensiones de matriculación que se precisen, y cada cuenta de usuario debe estar asociada a una de ellas.

Además de las extensiones incluidas por defecto, se pueden añadir extensiones de terceros o desarrollarlas a medida.

Evaluación de las extensiones nativas

Matriculación manual

La matriculación de usuarios corre a cargo de un administrador, profesor o gestor del curso.

Es una tarea inviable mantener actualizadas todas las matriculaciones de los alumnos y profesores en cada curso, ya que el número de usuarios supera con creces los 20.000 usuarios y el de asignaturas los 4.000.

Auto-matriculación

Los usuarios pueden matricularse por sí mismos en los cursos. Esto va en contra de los requisitos **R01 - R04**, que especifican que los usuarios sólo pueden estar en aquellos cursos en los que estén matriculados como alumnos o asignados como profesores.

Sincronización de cohorte

Las cohortes son grupos de usuarios a nivel de sitio Moodle. Con esta extensión, todos los usuarios pertenecientes a una cohorte se matriculan en un curso. No es aplicable en este modelo.

Metaenlace de curso

Los usuarios ya matriculados en un curso, se pueden matricular en otro. No es aplicable en este modelo.

Acceso de invitados

Cualquier usuario, sin matricularse en un curso, puede acceder al mismo. No cumple con los requisitos **R01 - R04**.

Categoría de matrícula

Una categoría engloba un conjunto de cursos. Utilizando esta extensión, los usuarios seleccionados se matriculan en todos los cursos de una categoría. No es aplicable en este modelo.

Base de datos externa

Esta extensión utiliza una base de datos externa para determinar qué usuarios deben participar en un curso y con qué rol.

Este proceso realiza las matriculaciones cuando el usuario se autentica. Si se desea que la actualización de matrículas sea un proceso automático, esta extensión cuenta con un script que se puede ejecutar regularmente para crear las nuevas cuentas, concordando con el requisito **R05**.

Esta extensión cumple con los requisitos especificados. Se estudiará con detenimiento para determinar si es válida sin realizar ninguna modificación.

Archivo plano (CSV)

La información de las matrículas de los usuarios se obtiene de un fichero de texto plano. Actualmente se utiliza una extensión basada en ésta, como se describe en el apartado *Estado actual*. Solucionar los problemas que surgen de su uso es una de las motivaciones de este proyecto.

Archivo IMS Enterprise

Se trata de un fichero de texto con un formato XML reconocido. Incurrir en la misma problemática que acarrea el uso de un fichero de texto plano, añadiendo dificultad en su generación y proceso.

Matriculaciones LDAP

El servidor LDAP de la ULPGC se utiliza únicamente a efectos de identificación, no conteniendo información relativa a las asignaturas en las que está matriculado un usuario, por lo que no se puede utilizar esta extensión, que se basa en la información almacenada en LDAP.

Matriculaciones remotas Mnet

Mnet define la conexión entre diferentes instalaciones de Moodle. Este mecanismo no se está utilizando en la ULPGC, por lo que se descarta el uso de ésta extensión.

Paypal

Este medio de pago no se utiliza en la ULPGC, descartando el uso de esta extensión, que efectúa la matrícula tras realizar un pago en Paypal.

Conclusión

De la evaluación anterior, teniendo en cuenta que la información de los usuarios se almacena en la base de datos institucional, procedemos a evaluar a fondo el funcionamiento de la extensión **Base de datos externa**. En el caso de que necesite alguna modificación, se desarrollará una nueva extensión.

Extensión de matriculación Base de datos a fondo

Son varias las bases de datos que se pueden utilizar con esta extensión, pero nos centraremos exclusivamente en Oracle, utilizada por la ULPGC como sistema gestor de base de datos. Asume la existencia de una tabla que contiene los siguientes campos:

- identificador de curso
- identificador de usuario
- rol del usuario

Matriculación y desmatriculación

La matriculación de un usuario tiene lugar cuando éste se autentica en Moodle. La extensión intenta automáticamente matricularlo en todos los cursos presentes en la base de datos externa y, opcionalmente, crear los cursos si no existen.

Este proceso también desmatricula a los usuarios de los cursos que ya no aparecen en la base de datos externa. Los registros de usuario se etiquetan con la extensión de matriculación utilizada. Por ello, la extensión sólo procederá a desmatricular a aquellos usuarios que hayan sido matriculados utilizando esta extensión en primer lugar.

Realizar el proceso de actualización de matrículas cada vez que accede un usuario supone un aumento significativo del tiempo de acceso a la plataforma. Como se mencionó anteriormente,

esta extensión dispone de un script para sincronizar automáticamente las matrículas. Utilizando éste, se evitará el uso de la extensión durante la autenticación.

Matriculación y roles

Aunque es posible especificar un rol por defecto en la página de configuración de la extensión, se utilizará un campo en la tabla de la base de datos externa para especificar el rol de cada usuario, dado que se van a matricular tanto estudiantes como profesores.

Creación de cursos

Opcionalmente, los cursos que no existen en Moodle se pueden crear, configurando adecuadamente la extensión, indicando incluso qué formato aplicar al curso y en qué categoría incluirlo. Sin embargo, es una de los requisitos, no de esta extensión sino del sistema en general, que los cursos sean creados con independencia de los usuarios que participen en ellos.

Script de sincronización

La extensión incluye un script que permite sincronizar todas las matrículas simultáneamente, tanto añadiendo como eliminando matrículas de usuarios. El script se llama *sync.php* y se encuentra en la carpeta *enrol/database/cli*. Este script está pensado para ejecutarse desde el cronjob del sistema. Los usuarios deben existir previamente en la plataforma, por lo que es necesario utilizar el script de sincronización similar de la extensión de autenticación utilizada.

Una entrada de ejemplo en el cron:

```
# 5 minutes past 4am
5 4 * * * /usr/bin/php -c /path/to/php.ini /path/to/moodle/enrol/database/cli/sync.php
```

Notas:

- Si el número de matrículas es muy grande, es conveniente elevar el límite de memoria con el argumento *-d memory_limit=256M*
- Para depurar y mantener un log, se recomienda añadir a la línea de comando *-d log_errors=1 -d error_reporting=E_ALL -d display_errors=0 -d html_errors=0*

Configurar la sincronización de usuarios

En primer lugar es necesaria una tabla en la base de datos externa que contenga un registro por cada matrícula de alumno o asignación de un profesor en una asignatura, con el rol del mismo.

Los campos necesarios son:

- Un identificador de curso único que coincida con uno de los siguientes campos en la tabla *course* de la base de datos de Moodle:
 - *idnumber* (varchar 100), correspondiente al Número ID del curso
 - *shortname* (varchar 255), nombre corto del curso
 - *id* (int 10), identificador autonumérico generado cuando se crea el curso
- Un identificador de usuario único, que debe coincidir con uno de los siguientes campos en la tabla *user* en la base de datos de Moodle:
 - *idnumber* (varchar 100), correspondiente al Número ID de usuario
 - *username* (varchar 100), nombre del usuario en Moodle
 - *email* (varchar 100), dirección de correo del usuario
 - *id* (int 10), identificador autonumérico generado cuando se crea el usuario
- Un identificador de rol único, que debe coincidir con uno de los siguientes campos en la tabla *role* en la base de datos de Moodle:
 - *shortname* (varchar 100), nombre corto del rol
 - *name* (varchar 255), nombre real del rol
 - *id*, identificador autonumérico generado cuando se crea el rol

En Moodle, activar la extensión en **Administración del sitio - Extensiones - Matriculaciones - Gestionar plugins de matriculación** y a continuación pulsar **Configuración**.

En la página de configuración hay que introducir los datos de conexión con la base de datos externa, así como la tabla en la misma con los datos de las matrículas y la correspondencia entre los campos de la base de datos externa y la base de datos de Moodle.

Hay varias opciones para configurar los roles por defecto en caso de que no se especifiquen en la base de datos externa, como gestionar la desmatriculación de usuarios y el manejo de los

cursos ocultos. En principio no son relevantes, salvo la desmatriculación, que trataremos más adelante.

Por último se puede configurar la creación automática de cursos.

Problemas potenciales

- La integridad de la base de datos externa es fundamental. Si los datos se pierden o no se generan correctamente en la base de datos externa, esta extensión por defecto desmatricula a usuarios de los cursos que no están en la tabla, eliminándolos de los grupos en los que estén asignados y borrando su actividad de los módulos que así lo especifiquen.
- Relacionado directamente con el apartado anterior, el hecho de que la anulación de matrículas se base en la ausencia de registros, fuerza a comprobar siempre la existencia de los mismos, perjudicando el rendimiento dado el alto número de matrículas gestionadas.
- Hay que tener en cuenta que los campos equivalentes entre ambas bases de datos no deben poder ser modificados en Moodle por los usuarios, puesto que podrían generar problemas de seguridad o incurrir en errores durante la sincronización. Por ejemplo, un usuario que modifica el valor de alguno de sus campos al valor del mismo campo de otro usuario válido, podría obtener acceso a cursos en los que está matriculado el usuario suplantado.

7.3.2.3 Resto de entidades a sincronizar

Para la gestión de centros, categorías, cursos y grupos no existe un esquema de extensiones, por lo que se desarrollarán extensiones a medida para sincronizar cada uno de estos ítems.

La creación de cursos y grupos de forma automática está integrada en la extensión de matriculación, pero en el ámbito de la solución actual es necesario que dispongan de una gestión independiente, puesto que la creación de cursos y grupos se realiza antes de matricular a los usuarios.

7.3.3 Descripción de los Interfaces entre Subsistemas

Para la comunicación entre ambos subsistemas se valoraron varias aproximaciones. A continuación explicamos brevemente los principales y, por último, nos centramos en el mecanismo escogido.

Ficheros de texto plano

Los ficheros de texto permiten exportar toda la información ya procesada desde la base de datos institucional. El tratamiento de la información se realiza en el SGBD, mucho más potente y rápido que posteriormente en el propio Moodle. En este solo sería necesario crear unas extensiones de identificación y matriculación que integrasen la información contenida en los ficheros de texto.

Para acelerar el proceso se utilizan ficheros planos en lugar de otras alternativas más potentes como podría ser XML.

Las principales ventajas de este sistema:

- la extracción y transformación de la información se realiza en el SGBD, no ralentizando el Campus Virtual y empleando un tiempo menor;
- sencillo de implementar;
- tanto las extensiones como los scripts utilizan un mecanismo similar para leer los ficheros y gestionar los datos;
- independencia en el tratamiento de la información y la carga de la misma.

Sin embargo, como se describió al inicio de esta sección, existen algunas desventajas que se consideran suficientes para descartar este mecanismo.

Conexión al LDAP institucional

La información de los usuarios se encuentra en un servidor LDAP externo. La ULPGC cuenta con un servidor LDAP contra el que se validan los usuarios vía el Servicio de Identificación Centralizada. Sin embargo, no contiene toda la información necesaria para crear una cuenta de usuario ni tampoco indica en qué titulaciones o asignaturas está matriculado un estudiante.

Uso de Servicios Web

La forma más elegante, directa y en tiempo real de sincronizar ambos sistemas. La versión 2 de Moodle incorpora una capa de Servicios Web muy potente que permite crear cursos, usuarios, realizar matrículas y otras muchas acciones. Sin embargo, las llamadas a estos Servicios Web deberían integrarse en las aplicaciones de gestión de los servicios de Gestión Académica y Ordenación Académica. Es un proyecto que requiere un estudio de viabilidad y una planificación e implantación minuciosas, a realizar paulatinamente, puesto que no se disponen ni de los recursos ni del tiempo suficiente para acometerlo en plazo.

Conexión directa a la base de datos corporativa

Estas extensiones consideran que los detalles de las cuentas de usuario y de su relación con las asignaturas están en una base de datos externa a Moodle, de la cual los obtiene y los inserta en la base de datos propia. Los principales inconvenientes:

- los procesos para obtener la información y procesarla son muy complejos, puesto que no hay una correspondencia directa entre ambas bases de datos; este problema es resuelto con el uso de vistas o tablas intermedias que faciliten la correspondencia directa.
- dada la gran cantidad de usuarios y cursos que se maneja, los recursos necesarios para obtener toda la información de una vez son insuficientes; si por el contrario se realizan varias conexiones, el riesgo de que fallen y el tiempo empleado aumentan considerablemente. Para solventar esta tara, se plantea la carga sistemática de la información diariamete, reduciendo considerablemente la cantidad de información a tratar.

Las principales ventajas de usar este mecanismo:

- sencillo de implementar: basta con instalar el cliente de la base de datos correspondiente para poder acceder directamente a su información;
- altamente configurable: la modificación de las consultas que obtienen la información de la base de datos externa es sencilla, pudiéndose parametrizar en el grado deseado.
- ejecución inmediata: no son necesarios pasos intermedios. Directamente se obtiene la información y se procesa.

7.4 ESPECIFICACIÓN DEL PLAN DE PRUEBAS

Para validar la adecuación del proyecto a los objetivos perseguidos, diseñamos un plan de pruebas de las extensiones y sus funciones, así como todos los mecanismos que utilizaremos para detectar errores y corregirlos ya en la fase de implementación.

Las pruebas contemplarán aspectos tanto de funcionalidad, ya que al ser un proceso automatizado no intervienen usuarios.

Se contemplarán tres tipos de pruebas:

- **Pruebas Unitarias:** serán útiles para probar el correcto funcionamiento de cada extensión, comprobando que cumple cada función concreta que implementa. Garantiza que cada una de las extensiones funciona correctamente por separado. A partir de los requisitos vistos anteriormente, se desarrollan pruebas unitarias y se especifican los resultados que se espera encontrar una vez ejecutada la operación sobre cada una de ellas
- **Pruebas de Integración:** comprenden verificaciones asociadas a grupos de funciones, verificando que todas trabajan correctamente cuando se combinan más de una extensión. Para ello, se probará cada escenario con varias entradas relevantes, incluyéndose entradas con datos correctos y con datos incorrectos para probar que el sistema reacciona correctamente ante errores.
- **Pruebas del sistema:** son pruebas de integración de todas las extensiones simultáneamente, que permiten probar el conjunto de todo el sistema y que sus relaciones son correctas, verificando así que todas sus especificaciones funcionales y técnicas se cumplen.

7.4.1 Casos de prueba

Cada escenario comprende los distintos casos de prueba relacionados con una categoría de requisitos (centros, usuarios, etc.). Cada caso representa una prueba unitaria, y la verificación del escenario completo es una prueba de integración. Comprobar la corrección de todos los escenarios simultáneamente equivale a una prueba del sistema.

<u>Escenario E1: Gestionar centros</u>	
<u>Caso de Prueba: CPI.1</u>	
Entrada	Resultado Esperado
Añadir un centro no existente	En la tabla de centros de la base de datos del Campus Virtual se crea una registro con la información del centro
<u>Caso de Prueba: CPI.2</u>	
Entrada	Resultado Esperado

Añadir un centro que ya existe	No se añade ningún registro nuevo en la tabla de centros de la base de datos del Campus Virtual y se mantiene el que ya existía
<i>Caso de Prueba: CP1.3</i>	
Entrada	Resultado Esperado
Eliminar centro	Se elimina el registro de la tabla de centros de la base de datos del Campus Virtual
<i>Caso de Prueba: CP1.4</i>	
Entrada	Resultado Esperado
Eliminar centro no existe	No se realiza ninguna modificación en el Campus Virtual

Escenario E2: Gestionar titulaciones

<i>Caso de Prueba: CP2.1</i>	
Entrada	Resultado Esperado
Añadir una titulación no existente	En el Campus Virtual se crea una categoría con el nombre de la titulación, dentro del centro que la oferta
<i>Caso de Prueba: CP2.2</i>	
Entrada	Resultado Esperado
Añadir una titulación que ya existe	No se añade ninguna categoría nueva y se mantiene la que ya existía
<i>Caso de Prueba: CP2.3</i>	
Entrada	Resultado Esperado

Eliminar una titulación	La categoría correspondiente a la titulación se elimina del Campus Virtual
<i>Caso de Prueba: CP2.4</i>	
Entrada	Resultado Esperado
Eliminar titulación no existe	No se realiza ninguna modificación en el Campus Virtual

<u>Escenario E3: Gestionar asignaturas</u>	
<i>Caso de Prueba: CP3.1</i>	
Entrada	Resultado Esperado
Añadir una asignatura no existente	En el Campus Virtual se crea un curso con el nombre de la asignatura, dentro de la titulación que la oferta
<i>Caso de Prueba: CP3.2</i>	
Entrada	Resultado Esperado
Añadir una asignatura que ya existe	No se añade ningún curso nuevo y se mantiene el existente
<i>Caso de Prueba: CP3.3</i>	
Entrada	Resultado Esperado
Eliminar asignatura	El curso correspondiente a la asignatura se elimina del Campus Virtual
<i>Caso de Prueba: CP3.4</i>	
Entrada	Resultado Esperado
Eliminar asignatura no existe	No se realiza ninguna modificación en el Campus Virtual

<u>Escenario E4: Gestionar grupos</u>	
<u>Caso de Prueba: CP4.1</u>	
Entrada	Resultado Esperado
Añadir un grupo no existente	En el Campus Virtual se crea un grupo en la asignatura correspondiente con el nombre del grupo
<u>Caso de Prueba: CP4.2</u>	
Entrada	Resultado Esperado
Añadir un grupo que ya existe	No se añade ningún grupo y se mantiene el que ya existía
<u>Caso de Prueba: CP4.3</u>	
Entrada	Resultado Esperado
Eliminar grupo	El grupo se elimina de la asignatura
<u>Caso de Prueba: CP4.4</u>	
Entrada	Resultado Esperado
Eliminar grupo no existe	No se realiza ninguna modificación en el Campus Virtual
<u>Caso de Prueba: CP4.5</u>	
Entrada	Resultado Esperado
Añadir grupo académico que se creó manualmente	Se modifica el grupo, marcándose como grupo académico
<u>Caso de Prueba: CP4.6</u>	
Entrada	Resultado Esperado

Asignar usuario matriculado en asignatura a grupo	El usuario está asignado al grupo en el curso correspondiente
<u>Caso de Prueba: CP4.7</u>	
Entrada	Resultado Esperado
Desasignar usuario matriculado en asignatura a grupo	El usuario no está asignado al grupo en el curso correspondiente
<u>Caso de Prueba: CP4.8</u>	
Entrada	Resultado Esperado
Asignar usuario matriculado en asignatura a grupo no existe	El usuario no está asignado a ningún grupo en el curso correspondiente

<u>Escenario E5: Gestionar usuarios</u>	
<u>Caso de Prueba: CP5.1</u>	
Entrada	Resultado Esperado
Añadir usuario que no existe	En el Campus Virtual se crea la cuenta del usuario
<u>Caso de Prueba: CP5.2</u>	
Entrada	Resultado Esperado
Añadir usuario que existe	No se realiza ninguna modificación en el Campus Virtual
<u>Caso de Prueba: CP5.3</u>	
Entrada	Resultado Esperado

Eliminar usuario que existe	Se elimina la cuenta del usuario en el Campus Virtual, además de desmatricularlo de todos los cursos en los que estuviese
<i>Caso de Prueba: CP5.4</i>	
Entrada	Resultado Esperado
Verificar acceso correcto	Acceder a una plataforma estando autenticado en CAS
<i>Caso de Prueba: CP5.5</i>	
Entrada	Resultado Esperado
Verificar restricción acceso	Acceder a una plataforma sin estar autenticado en CAS
<i>Caso de Prueba: CP5.6</i>	
Entrada	Resultado Esperado
Matricular usuario en curso	El usuario está matriculado en el curso correspondiente con el rol asignado
<i>Caso de Prueba: CP5.7</i>	
Entrada	Resultado Esperado
Desmatricular usuario de curso	El usuario no está matriculado en el curso correspondiente

8 DISEÑO DEL SISTEMA

8.1 ARQUITECTURA DE SISTEMAS

Como se describió en el apartado **4.4.3 Arquitectura de sistemas del Campus Virtual**, hay tres sistemas involucrados en el proceso de sincronización:

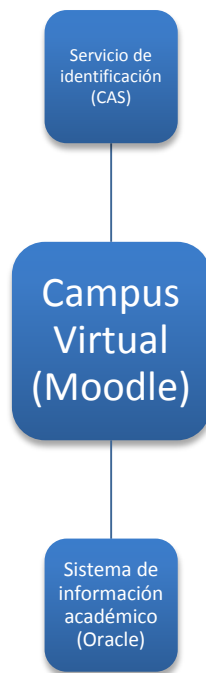


Figura 8-1 Sistemas involucrados

Realmente la sincronización se realiza con la información almacenada en el *Sistema de información académico*, pero la extensión de identificación tiene que comunicarse con el *Servicio de identificación* de la ULPGC para autenticar a los usuarios (requisito F13).

8.1.1 Comunicación con el Sistema de información académico

Como se estableció durante la fase de análisis, **7.3.3 Descripción de las interfaces entre subsistemas**, la conexión directa a la base de datos institucional es el mecanismo mejor valorado.

Moodle incorpora y utiliza la librería de abstracción de base de datos ADOdb. Se empleará la misma para conectarse con la base de datos institucional de la ULPGC.

Rendimiento. Se realizarán el menor número de consultas, a ser posible solo una, para reducir el tiempo de ejecución de cada script de sincronización.

Seguridad. La conexión entre ambos sistemas se realiza íntegramente en una red privada a la que no tiene acceso usuarios ajenos al Servicio de Informática.

8.1.2 Comunicación con el Sistema de identificación (CAS)

Moodle ya dispone de una extensión de identificación que utiliza CAS como servicio de identificación externo, haciendo uso del cliente oficial de CAS escrito en PHP, **phpCAS**. Se hará uso del mismo para implementar la nueva extensión de identificación.

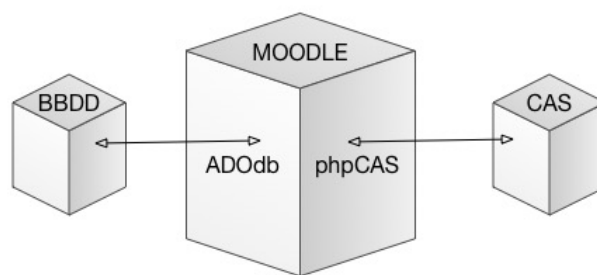


Figura 8-1 Comunicación entre sistemas

8.2 COMPONENTES DEL SISTEMA DE SINCRONIZACIÓN

Dada la estructura de Moodle y las entidades a sincronizar, este servicio se va a ofrecer empleando dos extensiones y cuatro scripts, en total 6 elementos para las 6 entidades a sincronizar.

Las entidades a sincronizar son: centros, titulaciones, asignaturas, grupos, usuarios (profesores y alumnos) y matrículas. Los usuarios y las matrículas se pueden sincronizar utilizando tipos de extensiones existentes en Moodle. Para el resto no hay definido ningún tipo de extensión, por lo que se utilizarán scripts en PHP, utilizando la API de Moodle.

Tanto los scripts como las extensiones son independientes entre sí, tanto por solicitud de los usuarios como por la propia arquitectura de Moodle: no existe ninguna relación ni interoperabilidad entre ellos. De hecho podrían utilizarse independientemente en diferentes instalaciones.

Sin embargo, es necesario matizar que sí existe una fuerte relación entre las diferentes entidades

En las últimas versiones de Moodle se tiende a que toda funcionalidad añadida tenga la forma de extensiones. En esta línea, los scripts de sincronización que no tienen un tipo de extensión equivalente se agruparán en una extensión de tipo **Local**.

En la siguiente figura se establece la correlación entre las entidades a sincronizar y el medio propuesto a tal fin.

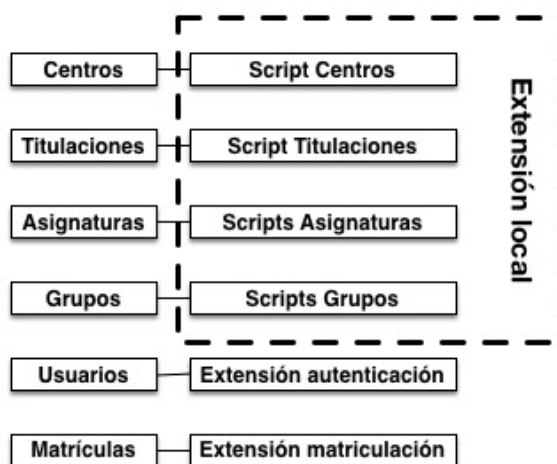


Figura 8-2 Relación de entidades a sincronizar y componentes del sistema

8.2.1 Elementos comunes a todas las extensiones

Configuración

Los valores de configuración del sistema de sincronización no se introducirán utilizando la interfaz de usuario, por motivos de seguridad.

En las distintas instalaciones de Moodle del Campus Virtual de la ULPGC hay varios administradores, con acceso global a todo el sistema. No es recomendable que puedan acceder y modificar esta extensión, ya que obtendrían información ajena a Moodle que podría facilitar el acceso al sistema de información de la ULPGC y podrían configurar incorrectamente el sistema de sincronización, con el siguiente malfuncionamiento del mismo.

Todo el código se versiona en Subversion salvo el fichero de configuración (*config.php*), por lo que si se añadiese un fichero de configuración local se podría acceder al mismo desde Subversion. Las dos opciones son añadir un fichero local y cambiar la política de uso de Subversion o añadir las variables de configuración al fichero *config.php*, utilizando la variable global *\$CFG*. Dado que esta última práctica ya se utiliza con otros fines, y solo las personas con

acceso físico a la máquina podrían ver este fichero, se opta por incluir los valores de configuración del sistema de sincronización en el fichero *config.php*.

Versión de la extensión

Para asegurar la compatibilidad de una extensión con la instalación de Moodle, así como para detectar cuándo es necesaria una actualización, se incluye un fichero en la carpeta de la extensión llamado *version.php*, un fichero estándar PHP, que empieza con la etiqueta `<?PHP`, no se cierra con la correspondiente etiqueta (`?>`), y define las siguientes variables:

- *\$plugin->version*
 - Obligatorio. Número de versión de la extensión con el formato *YYYYMMDDxx*, donde *xx* es el número de versión de ese día en concreto.
- *\$plugin->requires*
 - Opcional. Versión mínima de Moodle que requiere la extensión. Consultar la página <http://docs.moodle.org/dev/Releases> para identificar los números de versión de Moodle.
- *\$plugin->cron*
 - Opcional. Intervalo de tiempo entre las llamadas a la función *cron* de la extensión. Definido a 0 desactiva la ejecución de la misma.
- *\$plugin->component*
 - Opcional. Nombre *frankenstyle* de la extensión, muy recomendado (ver <http://docs.moodle.org/dev/Frankenstyle>). Utilizado para diagnóstico de instalación y actualización.
- *\$plugin->maturity*
 - Opcional. Estabilidad de la extensión: `MATURITY_ALPHA`, `MATURITY_BETA`, `MATURITY_RC`, `MATURITY_STABLE`
- *\$plugin->release*
 - Opcional. Número de versión en un formato simplificado, más sencillo de manejar.
- *\$plugin->dependencies*

- Opcional. Lista de otras extensiones que requiere este plugin para trabajar. Ejemplo: `array('mod_forum'=>ANY_VERSION, 'mod_data'=>2010020300)`.

Base de datos

Si la extensión utiliza nuevas tablas o modificar tablas en la base de datos, incluirá los ficheros *db/install.xml* y *db/upgrade.php* respectivamente:

- **db/install.xml**: fichero en formato xml describiendo las tablas y campos que requiere la extensión.
- **db/upgrade.php**: código para modificar las tablas de la extensión.

Idioma

Los paquetes de idiomas se incluyen en la carpeta *lang*. El paquete de idioma de un módulo es una carpeta con el nombre *xx_utf8*, incluyendo en el mismo una carpeta llamada *help* (contiene los ficheros de ayuda del módulo) y un archivo con el mismo nombre del módulo. *xx* es el código de dos caracteres del idioma, por ejemplo, *en* (inglés), *es* (español), *fr* (francés).

Scripts de línea de comandos

Los scripts que se ejecutan desde línea de comandos se almacenan en la carpeta **cli** dentro de la carpeta principal de la extensión.

8.2.2 Extensión local: sincronización de centros, titulaciones, asignaturas, grupos, asignación de grupos

Para la sincronización de los centros, titulaciones, asignaturas y grupos no existen extensiones propiamente definidas en Moodle. Hay algunas aproximaciones que permiten, por ejemplo, crear categorías y cursos (equivalente en Moodle a las titulaciones y asignaturas respectivamente) a partir de un archivo Excel, de poca utilidad en este caso.

Para estos scripts crearemos una extensión local, aquellas que no coinciden con ningún tipo de extensión definido.

Siguiendo la nomenclatura *Frankenstyle*, se llamará *local_sinculpgc* (al no coincidir con los tipos de extensión existentes se incluye en el directorio *local* de Moodle) y se almacenará en *local/sinculpgc*.

Diseño de la base de datos

Los centros no existen en Moodle. Para poder almacenarlos es necesario crear una nueva tabla. El uso de esta tabla en el entorno del Campus Virtual no compete a este proyecto, por lo que durante la realización del mismo no fue necesario establecer ninguna relación con el resto de tablas.

centres	
PK	<u>id</u>
	code name director secretary

Figura 8-2 Tabla de centros

Las tablas para el resto de entidades ya existen:

- **course_categories**: tabla de categorías. Las titulaciones en Moodle se crearán como categorías.
- **course**: tabla de cursos. Las asignaturas en Moodle se crearán como cursos.
- **groups**: tabla de grupos, donde se crearán los grupos de las asignaturas.
- **groups_members**: relación de alumnos asignados a los grupos de asignaturas.

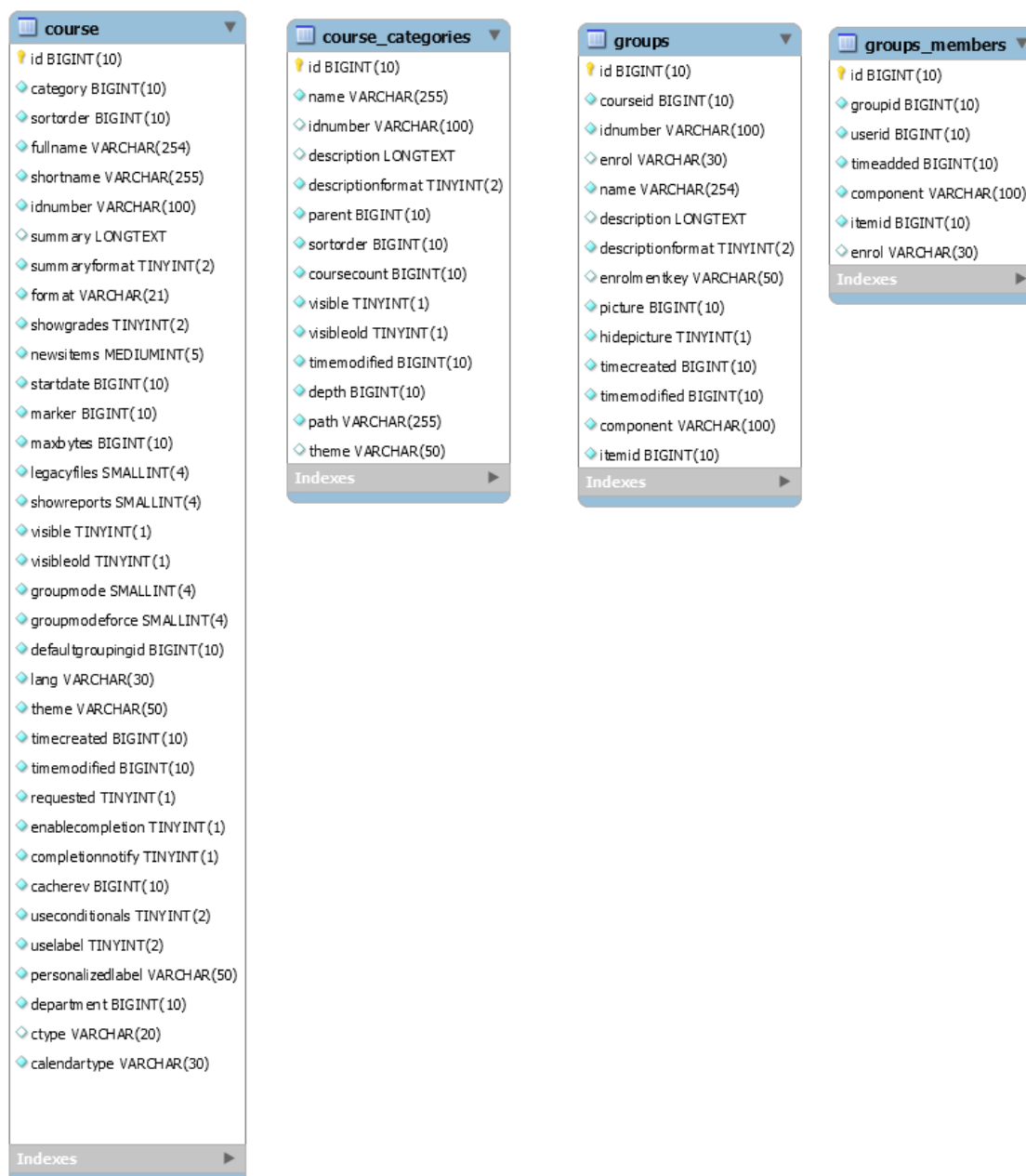


Figura 8-3 Tablas de Moodle implicadas en la sincronización

Utilizando la API de Moodle, en la mayoría de las ocasiones, la estructura de la base de datos es transparente para el desarrollador.

Diagrama de flujo

Los scripts para la creación de centros, titulaciones, asignaturas y grupos siguen un esquema similar al que se muestra en el siguiente diagrama de flujo. Los registros en la base de datos institucional indican si el registro se debe insertar, actualizar o eliminar, por lo que la acción a realizar ya está determinada antes de comenzar el proceso de sincronización.

Dado que en las plataformas del Campus Virtual se pueden realizar acciones sin que la base de datos institucional tenga constancia, por ejemplo, desmatricular a un alumno de un curso, siempre es necesario comparar la información en el Campus Virtual con la información en la base de datos institucional.

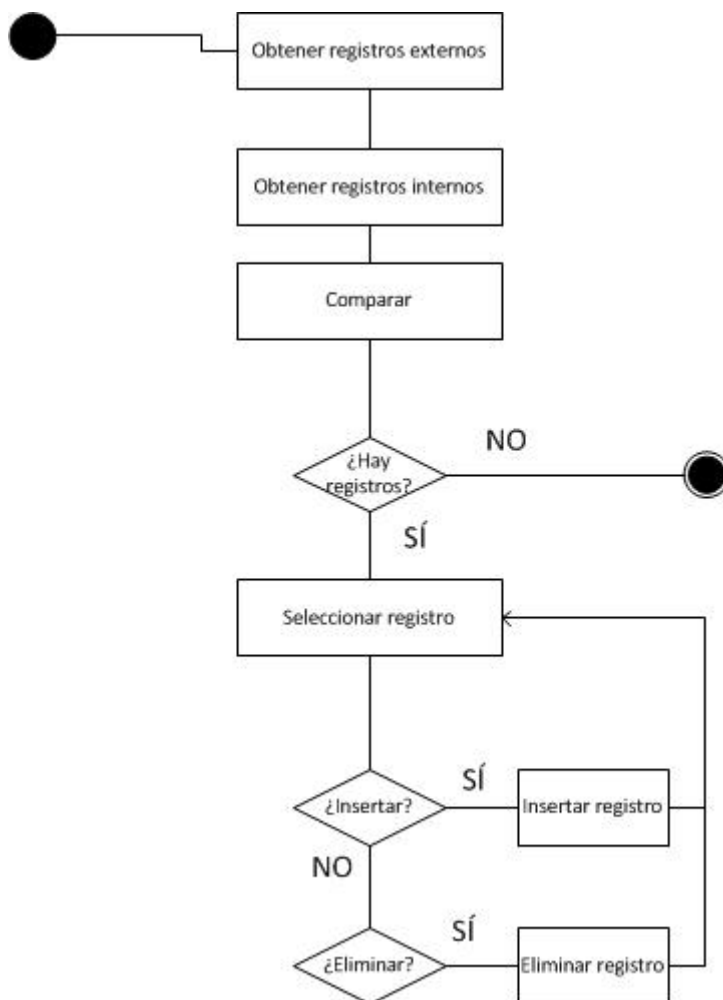


Figura 8-4 Diagrama de flujo de datos general scripts extensión local

Los procesos de insertar y eliminar no son similares, por lo que se desgranar en diferentes flujos de datos para cada entidad.

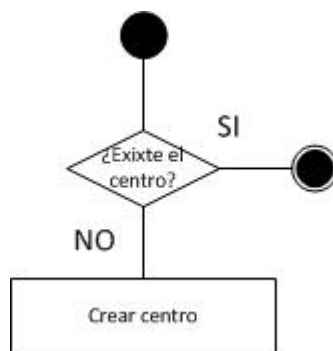


Figura 8-5 Diagrama de flujo de datos Insertar Centro

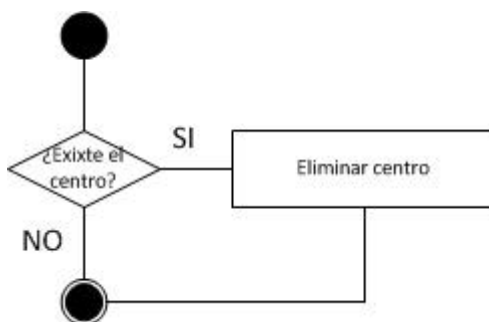


Figura 8-6 Diagrama de flujo de datos Eliminar Centro

Las entidades existentes en el Campus Virtual no se eliminan cuando no están en la base de datos institucional, sino que se ocultan para evitar pérdida de información en caso de tratarse de un error o volver a estar presentes en la base de datos institucional.

Por ejemplo, si una titulación se elimina en las aplicaciones corporativas por error, se eliminaría también del Campus Virtual junto con todas sus asignaturas. En lugar de esto, se oculta la categoría y los cursos que contiene, solo siendo visible a los administradores. De esta forma, al detectar el error y corregirlo en las aplicaciones de gestión, en el Campus Virtual solo es necesario volver a poner visible la categoría correspondiente a la titulación, así como los cursos, y no habría pérdida de información.

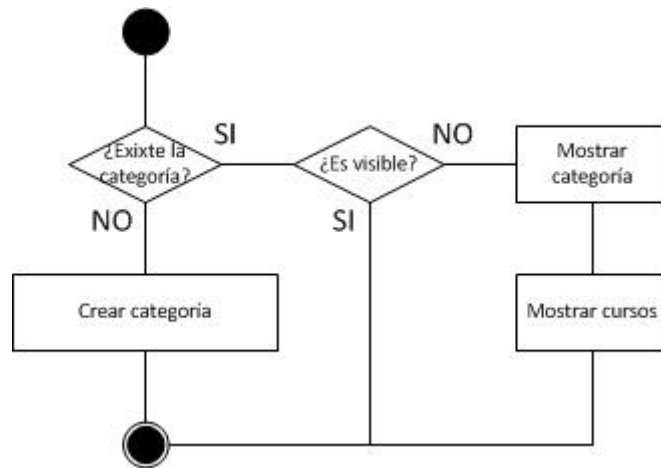


Figura 8-7 Diagrama de flujo de datos Insertar Categoría

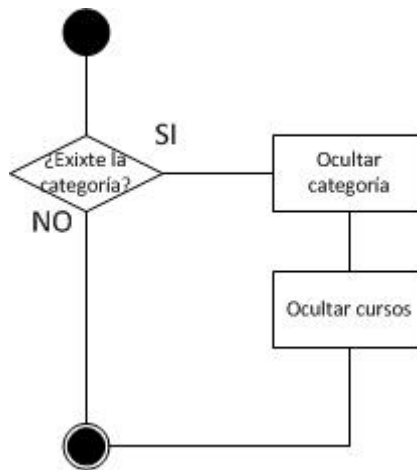


Figura 8-8 Diagrama de flujo de datos Eliminar Categoría

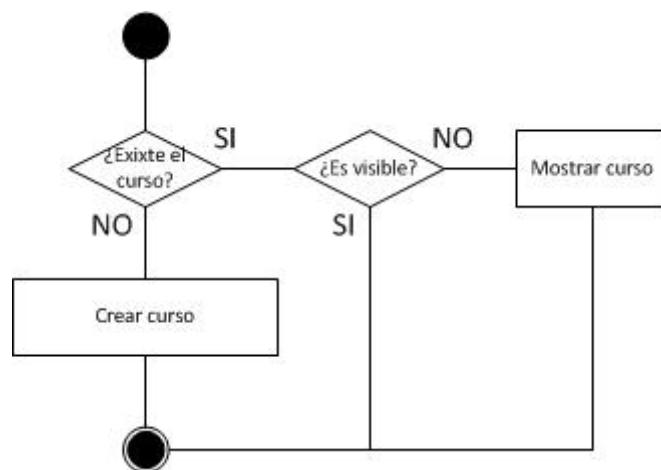


Figura 8-9 Diagrama de flujo de datos Insertar Curso

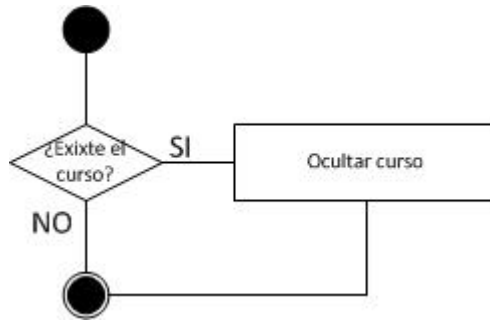


Figura 8-10 Diagrama de flujo de datos Eliminar Curso

En el Campus Virtual los profesores pueden crear grupos propios dentro de sus asignaturas. Para diferenciar los grupos propios de los grupos académicos, estos últimos se definen grupos ULPGC. En el caso de que un grupo ULPGC ya existe como propio, el grupo propio se redefine como grupo ULPGC. Esta diferenciación es necesaria porque los grupos ULPGC no pueden ser modificados por los profesores, cosa que no ocurre con los grupos que crean personalmente.

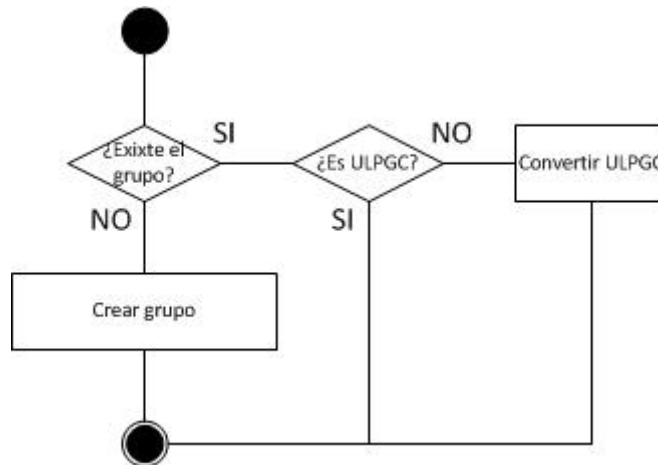


Figura 8-11 Diagrama de flujo de datos Insertar Grupo

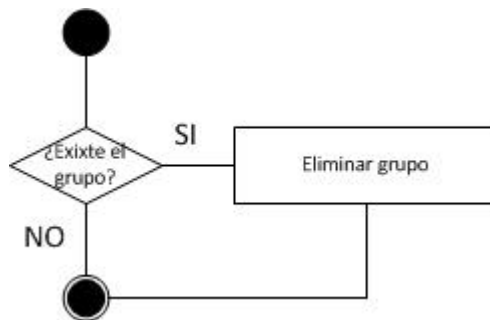


Figura 8-12 Diagrama de flujo de datos Eliminar Grupo

La asignación de grupo a los usuarios difiere ligeramente de las anteriores, al ser un proceso más complejo.

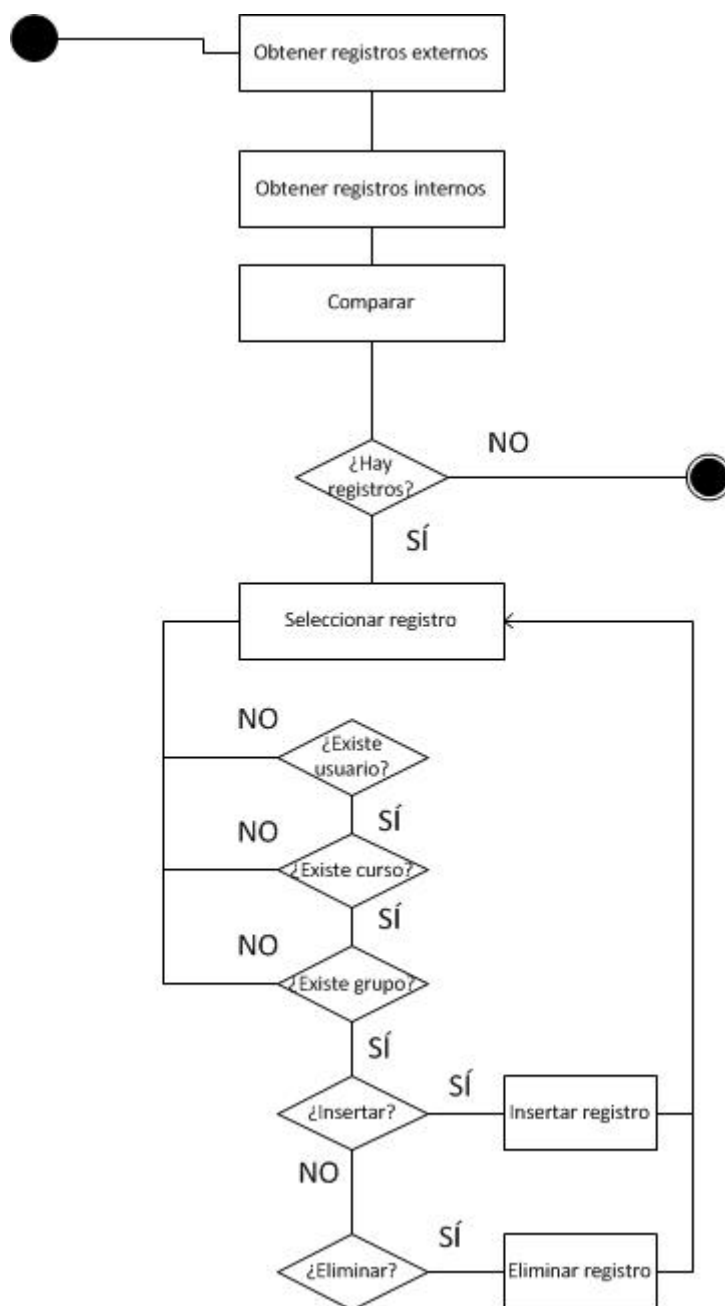


Figura 8-13 Diagrama de flujo de datos Asignación Grupos

Al igual que sucede con los grupos, la asignación de grupos la pueden realizar manualmente los profesores o gestores de un curso, así como eliminarla. Las asignaciones realizadas a nivel académico no se pueden modificar por los usuarios del Campus Virtual, por lo que se les categoriza como provenientes de la ULPGC para impedir su tratamiento.

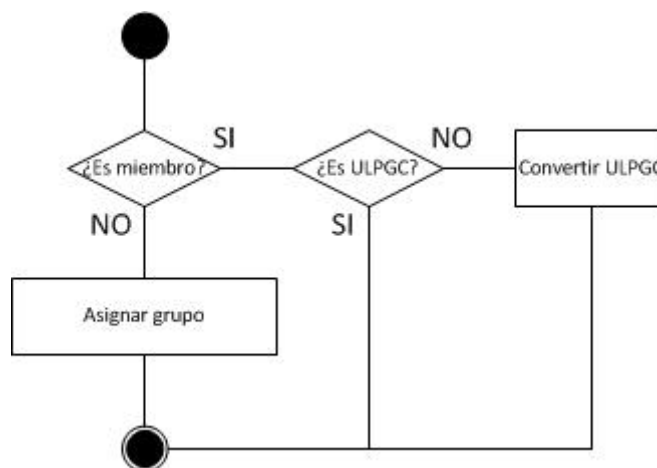


Figura 8-14 Diagrama de flujo de datos Asignar Grupo

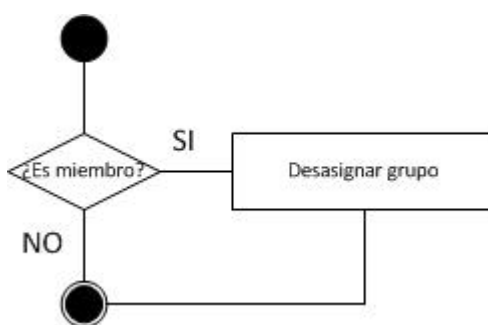


Figura 8-15 Diagrama de flujo de datos Desasignar Grupo

Todos los scripts se almacenarán en la carpeta **cli** de la extensión, **local/sinculpgc/cli**, puesto que se ejecutan desde línea de comandos.

8.2.3 Extensión de identificación

A partir de la documentación de Moodle y las extensiones existentes, principalmente la que permite utilizar una base de datos externa, analicemos los distintos aspectos a tener en cuenta para crear una extensión de identificación.

Proceso de autenticación

El proceso de autenticación empieza cuando un usuario pulsa sobre el enlace Entrar en el sitio Moodle. A continuación:

1. Se muestra la página de autenticación por defecto. Para utilizar otra página, es necesario asignar la URL de la misma a la opción de configuración URL de acceso alternativo (*Administración del sitio - Extensiones - Identificación - Gestionar identificación*).
2. El usuario introduce sus credenciales y envía el formulario.

3. El código del manejador en `/login/index.php` gestiona el proceso:
 - a. Obtiene una lista de las extensiones de autenticación habilitadas.
 - b. Lanza la función `loginpage_hook()` de cada extensión, en caso de que alguna de ellas necesite actuar sobre la petición de autenticación.
 - c. Comprueba que el nombre de usuario es válido, según los criterios establecidos en Moodle.
 - d. Llama la función `authenticate_user_login()` en `/lib/moodlelib.php`, que devuelve un objeto `$user`.
 - e. Determina si la autenticación tuvo éxito (si `$user` es un objeto válido). En caso afirmativo, muestra al usuario la página solicitada inicialmente o, por el contrario, le muestra la página de autenticación con un mensaje de error.

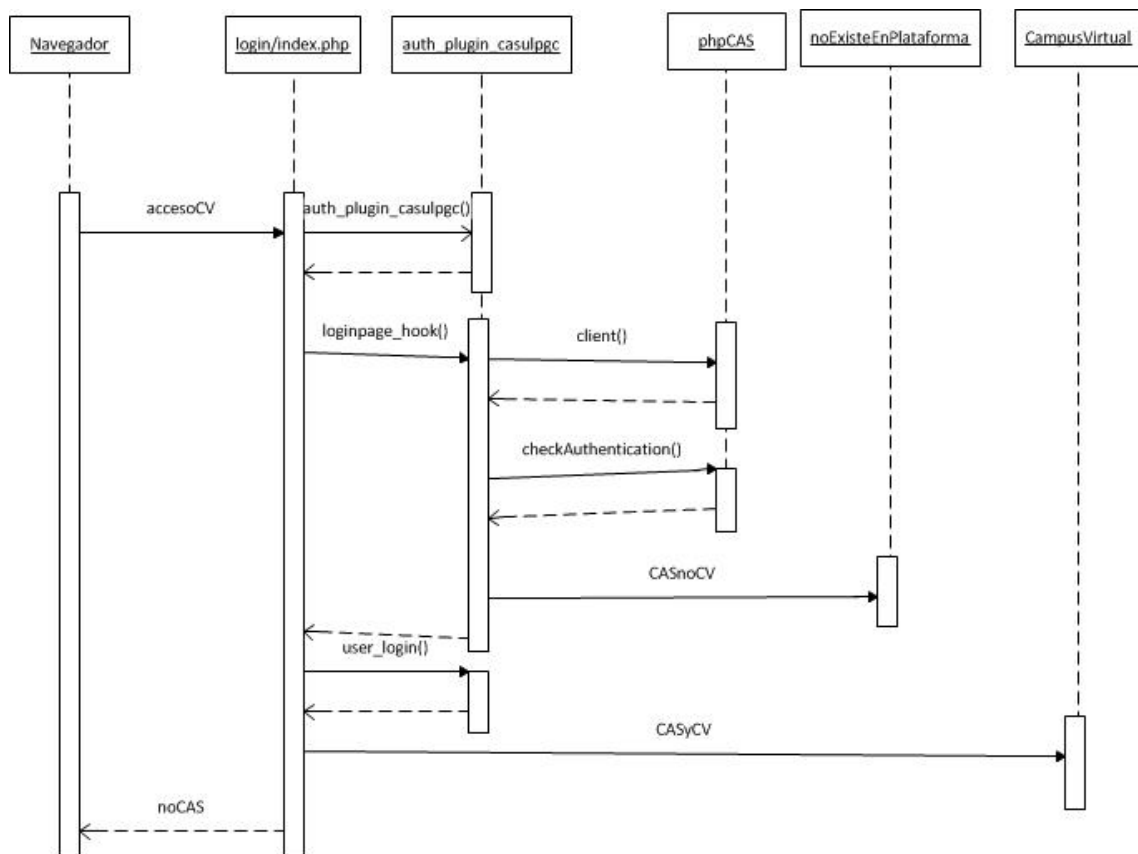


Figura 8-16 Diagrama de secuencia del proceso de identificación

Al intentar acceder a una plataforma del Campus Virtual se comprueba si existe una sesión abierta en el CAS corporativo. Si es así, pueden darse dos casos:

- el usuario que tiene sesión abierta en CAS no tiene acceso a la plataforma, redirigiéndosele a una página indicado esta situación;
- el usuario que tiene sesión abierta tiene acceso a la plataforma, redirigiéndolo a la misma.

Si no existe sesión abierta en CAS, el usuario es redirigido a la página de identificación del Servicio de identificación centralizada.

Estructura de la extensión

1. En la instalación de Moodle, hay que crear un directorio en */auth* con el nombre de la nueva extensión, a la que denominaremos **casulpgc**. Creamos la carpeta */auth/casulpgc*.
2. Hay que crear el fichero */auth/casulpgc/auth.php*. En este fichero creamos la clase *auth_plugin_casulpgc*, que extiende la clase *auth_plugin_base*, definida en */lib/authlib.php*.

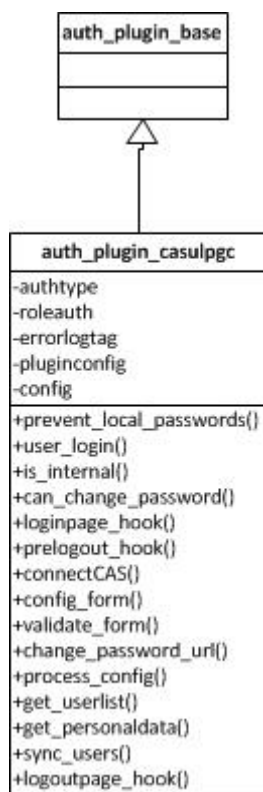


Figura 8-17 Diagrama de clases extensión Identificación

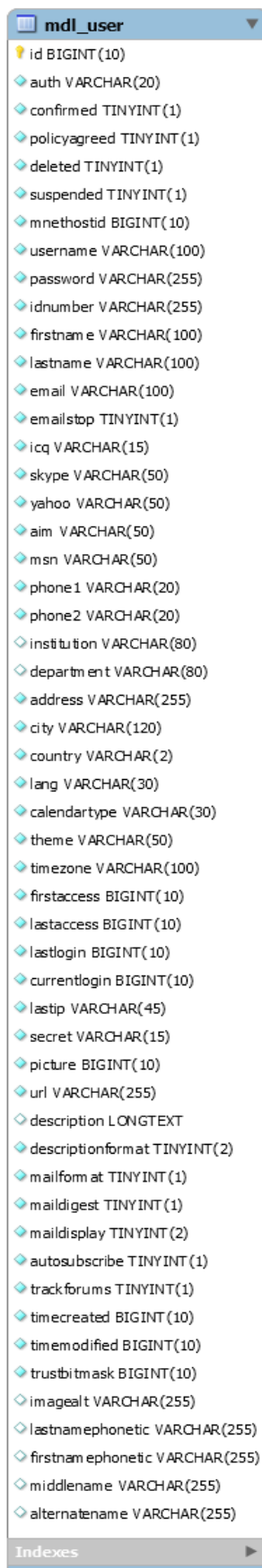
3. Implementamos la función *user_login()* en *lib/authlib.php*, creando o sobrescribiendo funciones adicionales según los requisitos de nuestra extensión. El resto de métodos y atributos modificados o nuevos están reflejados en el diagrama de clases.

4. Para configurar la extensión utilizando la interfaz de usuario, hay que implementar los métodos *config_form()* y *process_config()*. A los ajustes de configuración se podrá acceder desde la página (**Administración del sitio - Extensiones - Identificación - CASULPGC**), siendo los valores de configuración almacenados en la tabla *config_plugins*.

La página de configuración es un fichero en formato html que se debe incluir en el directorio de la extensión, con el nombre *config.html*. Básicamente será un formulario con las opciones de configuración de la extensión.

Base de datos

La información de los usuarios, conteniendo tanto lo datos personales como la extensión de identificación que utilizan, se almacena en la tabla **user**. La interacción con esta tabla se realiza íntegramente utilizan la API de Moodle.



The image shows a screenshot of a database schema viewer for a table named 'mdl_user'. The table is listed with its columns and their data types. The columns are: id (BIGINT(10)), auth (VARCHAR(20)), confirmed (TINYINT(1)), policyagreed (TINYINT(1)), deleted (TINYINT(1)), suspended (TINYINT(1)), mnehtostid (BIGINT(10)), username (VARCHAR(100)), password (VARCHAR(255)), idnumber (VARCHAR(255)), firstname (VARCHAR(100)), lastname (VARCHAR(100)), email (VARCHAR(100)), emailstop (TINYINT(1)), icq (VARCHAR(15)), skype (VARCHAR(50)), yahoo (VARCHAR(50)), aim (VARCHAR(50)), msn (VARCHAR(50)), phone1 (VARCHAR(20)), phone2 (VARCHAR(20)), institution (VARCHAR(80)), department (VARCHAR(80)), address (VARCHAR(255)), city (VARCHAR(120)), country (VARCHAR(2)), lang (VARCHAR(30)), calendartype (VARCHAR(30)), theme (VARCHAR(50)), timezone (VARCHAR(100)), firstaccess (BIGINT(10)), lastaccess (BIGINT(10)), lastlogin (BIGINT(10)), currentlogin (BIGINT(10)), lastip (VARCHAR(45)), secret (VARCHAR(15)), picture (BIGINT(10)), url (VARCHAR(255)), description (LONGTEXT), descriptionformat (TINYINT(2)), mailformat (TINYINT(1)), maildigest (TINYINT(1)), maildisplay (TINYINT(2)), autosubscribe (TINYINT(1)), trackforums (TINYINT(1)), timecreated (BIGINT(10)), timemodified (BIGINT(10)), trustbitmask (BIGINT(10)), imagealt (VARCHAR(255)), lastnamephonetic (VARCHAR(255)), firstnamephonetic (VARCHAR(255)), middlename (VARCHAR(255)), and alternatename (VARCHAR(255)). At the bottom of the window, there is a section for 'Indexes' with a right-pointing arrow.

Column Name	Data Type
id	BIGINT(10)
auth	VARCHAR(20)
confirmed	TINYINT(1)
policyagreed	TINYINT(1)
deleted	TINYINT(1)
suspended	TINYINT(1)
mnehtostid	BIGINT(10)
username	VARCHAR(100)
password	VARCHAR(255)
idnumber	VARCHAR(255)
firstname	VARCHAR(100)
lastname	VARCHAR(100)
email	VARCHAR(100)
emailstop	TINYINT(1)
icq	VARCHAR(15)
skype	VARCHAR(50)
yahoo	VARCHAR(50)
aim	VARCHAR(50)
msn	VARCHAR(50)
phone1	VARCHAR(20)
phone2	VARCHAR(20)
institution	VARCHAR(80)
department	VARCHAR(80)
address	VARCHAR(255)
city	VARCHAR(120)
country	VARCHAR(2)
lang	VARCHAR(30)
calendartype	VARCHAR(30)
theme	VARCHAR(50)
timezone	VARCHAR(100)
firstaccess	BIGINT(10)
lastaccess	BIGINT(10)
lastlogin	BIGINT(10)
currentlogin	BIGINT(10)
lastip	VARCHAR(45)
secret	VARCHAR(15)
picture	BIGINT(10)
url	VARCHAR(255)
description	LONGTEXT
descriptionformat	TINYINT(2)
mailformat	TINYINT(1)
maildigest	TINYINT(1)
maildisplay	TINYINT(2)
autosubscribe	TINYINT(1)
trackforums	TINYINT(1)
timecreated	BIGINT(10)
timemodified	BIGINT(10)
trustbitmask	BIGINT(10)
imagealt	VARCHAR(255)
lastnamephonetic	VARCHAR(255)
firstnamephonetic	VARCHAR(255)
middlename	VARCHAR(255)
alternatename	VARCHAR(255)

Figura 8-18 Tabla de usuarios de Moodle

Localización de la extensión

Por defecto, nuestro plugin aparecerá como `[[auth_casulpgc]]`. La asignación de nombres a las identificaciones se realiza creando un fichero de idioma para la extensión. Creamos el directorio `/auth/casulpgc/lang/` y dentro de éste un directorio por cada idioma que se precise, por ejemplo, `/auth/casulpgc/lang/es`. Dentro de estos directorios se crea un archivo `auth_casulpgc.php`, en el que se asigna la cadena que queremos se muestre en el menú anterior a la variable `$string['pluginname']`. La descripción de la extensión se asigna a `$string['auth_casulpgcdescription']`. Hay definidas otra serie de variables a las que se puede asignar la cadena de texto en el idioma adecuado.

8.2.4 Extensión de matriculación

Para corregir los problemas descritos en el análisis detallado de la extensión Base de datos, cuya funcionalidad cubre en buena medida los requisitos especificados, desarrollamos una nueva extensión. No se realizan las modificaciones sobre la misma para evitar problemas cuando se realicen actualizaciones de código o cambios de versión.

La matrícula da a los usuarios los siguientes privilegios:

- Un usuario con matrícula activa puede acceder a un curso. Los usuarios que no tienen, requieren acceso de invitado o el permiso `moodle/course:view`.
- La página “Mis cursos” muestra una lista de todas las matrículas activas del usuario.
- Participación en el curso - algunas actividades restringen la participación a usuarios matriculados exclusivamente.
- Solo los usuarios matriculados pueden ser miembros de grupos.
- El libro de calificaciones contempla a los alumnos matriculados, controlándose por el rol de los usuarios la visibilidad de las calificaciones.

Las matrículas y la asignación de roles son conceptos separados. Es posible estar matriculado en un curso y no tener ningún rol asignado y viceversa. Los roles a nivel de curso y por debajo se pueden controlar por las extensiones de matriculación.

Estructura de la extensión

Todas las extensiones de matriculación deben extender la clase *enrol_plugin*, definida en *lib/enrollib.php*, que contiene todos los métodos estándar.

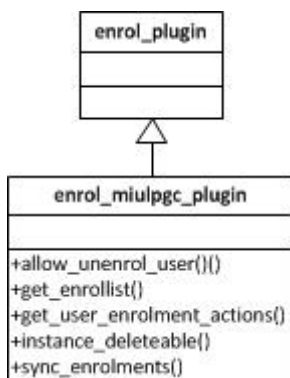


Figura 8-19 Diagrama de clases extensión de matriculación

Base de datos

La información de las matrículas de los cursos se almacena en las tablas *enrol* y *user_enrolments*, estando el rol asociado en la tabla *role_assignments*. Opcionalmente se utilizan otras tablas definidas por diferentes extensiones de matriculación. Cada extensión controla todos sus registros y las matrículas de los usuarios; por defecto, las matrículas de usuario están protegidas y no pueden ser modificadas manualmente por los profesores.



Figura 8-20 Tablas relacionadas con la matriculación

Suspensión y expiración de una matrícula

Una matrícula se considerará activa si cumple todas las condiciones siguientes:

- existe un registro para el usuario en la tabla *user_enrolments*,
- la matrícula está vigente, es decir, entre las fechas *enrolstartdate* y *enrolenddate*,
- la matrícula del usuario está activa (*status*)
- el tipo de matrícula está activa en la tabla *enrol* (campo *status*)
- la extensión de matriculación está activada

8.2.5 Secuencia de ejecución

Existe una clara dependencia entre las diferentes entidades, por lo que es forzoso seguir un orden en el proceso de sincronización. No sería lógico intentar matricular alumnos en un curso que todavía no se ha creado o intentar asignar un grupo en una asignatura a un alumno que no se ha matriculado. El script que automatice el proceso debe tener en cuenta esta restricción.

Hay que matizar que la ejecución de los scripts o el uso de las extensiones si es autónomo, su funcionamiento será totalmente correcto aunque se ejecuten de forma independiente. Pero los resultados pueden no ser los esperados sin no existen los elementos correspondientes. Para crear un grupo tiene que existir el curso, para asignar un usuario a un grupo tiene que existir previamente, etc.



Figura 8-21 Secuencia de ejecución de los scripts de sincronización

8.3 ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS

El proceso de pruebas se extiende durante todo el proceso de construcción del software. En esta sección se describe cómo se van a aplicar las pruebas diseñadas.

El entorno en el que se ha realizado el desarrollo es similar en prestaciones al entorno de producción. La principal diferencia es que no hay balanceo de carga, pero en este caso no es significativo porque el proceso de sincronización se ejecutará en un solo nodo, no existiendo en ningún caso balanceo de carga.

8.3.1 Pruebas Unitarias

Para la realización de las pruebas descritas en el apartado **7.4.1 Casos de prueba**, se crearán los registros pertinentes en la base de datos desde la que se extrae la información. Al ser una base de datos dedicada a la sincronización del Campus Virtual, es posible insertar, eliminar y modificar registros con total libertad. Hay un mecanismo que generará la información real a sincronizar en un corto periodo de tiempo, acción que se llevará a cabo una vez finalizadas las pruebas y verificado el correcto funcionamiento del proyecto.

Una vez implementado cada uno de los scripts y extensiones, se realizarán las pruebas correspondientes a cada uno de ellos. Para reutilizar los mismos casos de prueba, se reseteará el estado de Moodle al estado inicial.

El proceso de realización de las pruebas se resume en los siguientes pasos:

1. Implementación del script o extensión.
2. Preparar base de datos institucional con los registros necesarios.
3. Ejecutar el script o extensión.
4. Verificar en la plataforma que los resultados son los esperados.
5. Pasar al siguiente script o extensión.

Notar que, como se indicó en el apartado **8.2.5 Secuencia de ejecución**, la ejecución de los scripts y/o extensiones durante el proceso de pruebas tiene que seguir el orden establecido, debido a la dependencia existente entre los diferentes elementos que se crean en la plataforma (un usuario debe existir antes de poder matricularse).

Centros

Se necesitan los siguientes registros en la vista de centros (VMOCENTROS):

- varios registros de inserción de centros que no existen en Moodle;
- un registro de inserción de un centro que ya existe en Moodle, para verificar que no se inserta dos veces;
- un registro de eliminación de un centro que no existe, transparente para el sistema;
- un registro de eliminación de un centro previamente creado en Moodle.

Categorías

Se necesitan los siguientes registros en la tabla de titulaciones (TMO CATEGORIAS):

- varios registros de inserción de titulaciones que no existen en Moodle;
- un registro de inserción de una titulación que ya existe en Moodle, para verificar que no se inserta dos veces;
- un registro de eliminación de una titulación que no existe en Moodle, transparente para el sistema;
- un registro de eliminación de una titulación previamente creada en Moodle.

Cursos

Se necesitan los siguientes registros en la tabla de asignaturas (TMO CURSOS):

- varios registros de inserción de asignaturas que no existen en Moodle;
- un registro de inserción de una asignatura que ya existe en Moodle , para verificar que no se inserta dos veces;
- un registro de eliminación de una asignatura que no existe, transparente para el sistema;
- un registro de eliminación de una asignatura previamente creada en Moodle.

Grupos

Se necesitan los siguientes registros en la tabla de grupos (TMO GRUPOS):

- varios registros de inserción de grupos que no existen en Moodle;
- un registro de inserción de un grupo que ya existe en Moodle, para verificar que no se inserta dos veces;
- un registro de inserción de un grupo creado en Moodle manualmente, para verificar que se convierte en grupo académico;

- un registro de eliminación de un grupo que no existe, transparente para el sistema;
- un registro de eliminación de un centro previamente creado en Moodle.

Asignaciones de grupo

Se necesitan los siguientes registros en la tabla de asignación de grupos (TMOGRUPOSUSUARIOS):

- varios registros de inserción de asignaciones no realizadas en Moodle;
- un registro de eliminación de una asignación previamente creada en Moodle;
- un registro de inserción de asignación en un grupo que no existe en un grupo, transparente para el sistema.

Extensión de identificación

Se necesitan los siguientes registros en la tabla de usuarios (TMOUSUARIOS):

- varios registros de inserción de usuarios que no existen en Moodle;
- un registro de inserción de un usuario que ya existe en Moodle, procedente de sincronización;
- un registro de inserción de un usuario que ya existe en Moodle, creado manualmente;
- un registro de eliminación de un usuario que existe en Moodle, procedente de sincronización;
- un registro de eliminación de un usuario que existe en Moodle, creado manualmente.

Extensión de matriculación

Se necesitan los siguientes registros en la tabla de matrículas (TMOMATRICULAS):

- varios registros de inserción de matrículas que no existen en Moodle;
- un registro de inserción de una matrícula que ya existe en Moodle, procedente de sincronización;
- un registro de inserción de una matrícula que ya existe en Moodle, creada manualmente;
- un registro de eliminación de una matrícula que existe en Moodle, procedente de sincronización;
- un registro de eliminación de una matrícula que existe en Moodle, creada manualmente.

8.3.2 Pruebas de Integración y del Sistema

La ejecución de todos los escenarios anteriores conjuntamente, no uno a uno, sino en bloque, ya supone una prueba de integración y de sistema, puesto que los scripts se integran en Moodle desde el inicio, para poder utilizar el esquema de extensiones y la API, y la dependencia entre los diferentes elementos fuerza que no se puedan probar de forma totalmente independiente, aunque, como se mencionó anteriormente, se pueden ejecutar por separado con éxito.

9 IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se recoge información relevante sobre la implementación del sistema, detalles que pueden facilitar la comprensión de la solución desarrollada. En el soporte digital que acompaña esta memoria está disponible todo el código fuente correspondientemente comentado.

9.1 PREPARACIÓN DEL ENTORNO

El Servicio de Informática de la ULPGC facilita un entorno similar al de producción del Campus Virtual. Se solicita la instalación del cliente Oracle y habilitar conexión directa desde servidor donde se encuentra instalado el código fuente de Moodle a la base de datos institucional. En el entorno de producción solo se utilizará un frontal para la sincronización, por lo que no es necesario disponer de un sistema redundante con balanceo de carga.

Como ya se describió con detalle en los apartados **4.4 Arquitectura física del Campus Virtual de la ULPGC** y **6.1 Arquitectura de Moodle**, Moodle precisa de un servidor web, una base de datos y un directorio de datos, que pueden estar en una sola máquina o en diferentes servidores, como es el caso, para mejorar el rendimiento. En el caso de la ULPGC, para la identificación de los usuarios hay que habilitar la conexión con un servidor CAS y, para la sincronización, con la base de datos institucional.

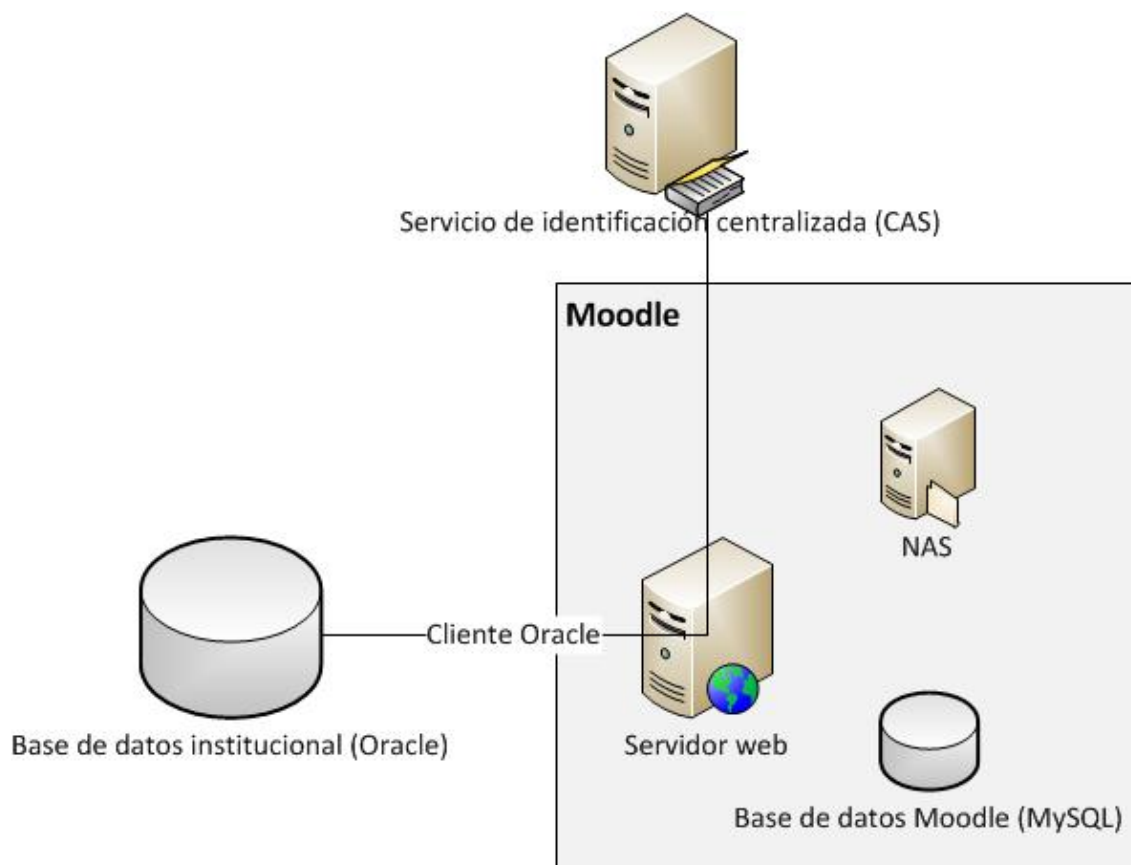


Figura 9-1 Entorno de desarrollo

9.2 CONEXIÓN A BASE DE DATOS INSTITUCIONAL

En el apartado **8.4.1 Elementos comunes a todas las extensiones** se adelantó la ubicación y los motivos por los valores de configuración del acceso a la base de datos institucional se incluirían en el fichero de configuración global, *config.php*.

En el citado fichero se añaden las siguientes variables, con los valores correspondientes que, por motivos de seguridad, no se incluyen en esta memoria:

```
$CFG->ulpgcdbtype = 'oci8';  
$CFG->ulpgcdbhost = 'dirección.bbdd.institucional:puerto';  
$CFG->ulpgcdbuser = 'USUARIO';  
$CFG->ulpgcdbpass = 'CONTRASEÑA';  
$CFG->ulpgcdbname = 'SERVICENAMEBDDORACLE';  
$CFG->ulpgccharset= 'we8iso8859p1';
```

Los valores de estas variables son los que utiliza el cliente Oracle, mediante la librería *ADODB*, para conectarse a la base de datos institucional.

La conexión con la base de datos externa se habilita con una serie de funciones definidas en el archivo *local/sinculpgc/lib.php*. El código fuente está correctamente comentado para facilitar su interpretación.

La conexión a la base de datos sigue el patrón de la siguiente figura:

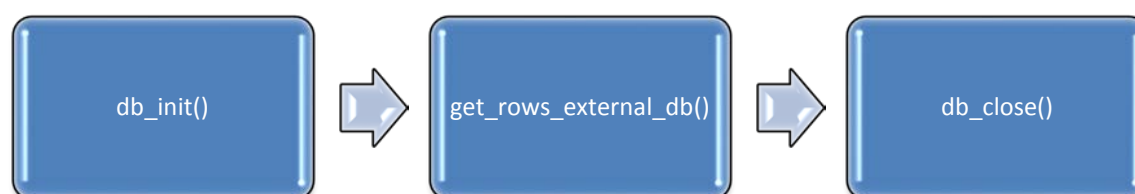


Figura 9-2 Conexión a base de datos externa

Básicamente, se establece la conexión con la base de datos externa, se obtienen los registros necesarios y se cierra la conexión. Internamente se utiliza una función para convertir el contenido de los registros a UTF-8, puesto que la base de datos institucional trabaja en ISO-8859-1 y el Campus Virtual en UTF-8.

9.3 AUDITORÍA

Aunque no se ha requerido por los usuarios, a efectos de controlar las modificaciones realizadas por el proceso de sincronización, se registran todos los movimientos que se realizan.

Se definen una serie de contadores, con valor estadístico, del número de registros que se tratan en cada ejecución del proceso, y, además de información individual de cada operación, se almacenan en la tabla **log**, empleando la API de *logging* de Moodle:

```
add_to_log($courseid, $module, $action, $url='', $info='', $cm=0, $user=0)
```

9.4 EXTENSIÓN LOCAL

Siguiendo la estructura de extensiones de Moodle, al no existir un tipo de extensión predefinido, la extensión de sincronización para centros, categorías, cursos, grupos y asignación de grupos se crea como una extensión local.

9.4.1 Estructura de archivos

La estructura de archivos de la extensión se muestra en la siguiente figura:

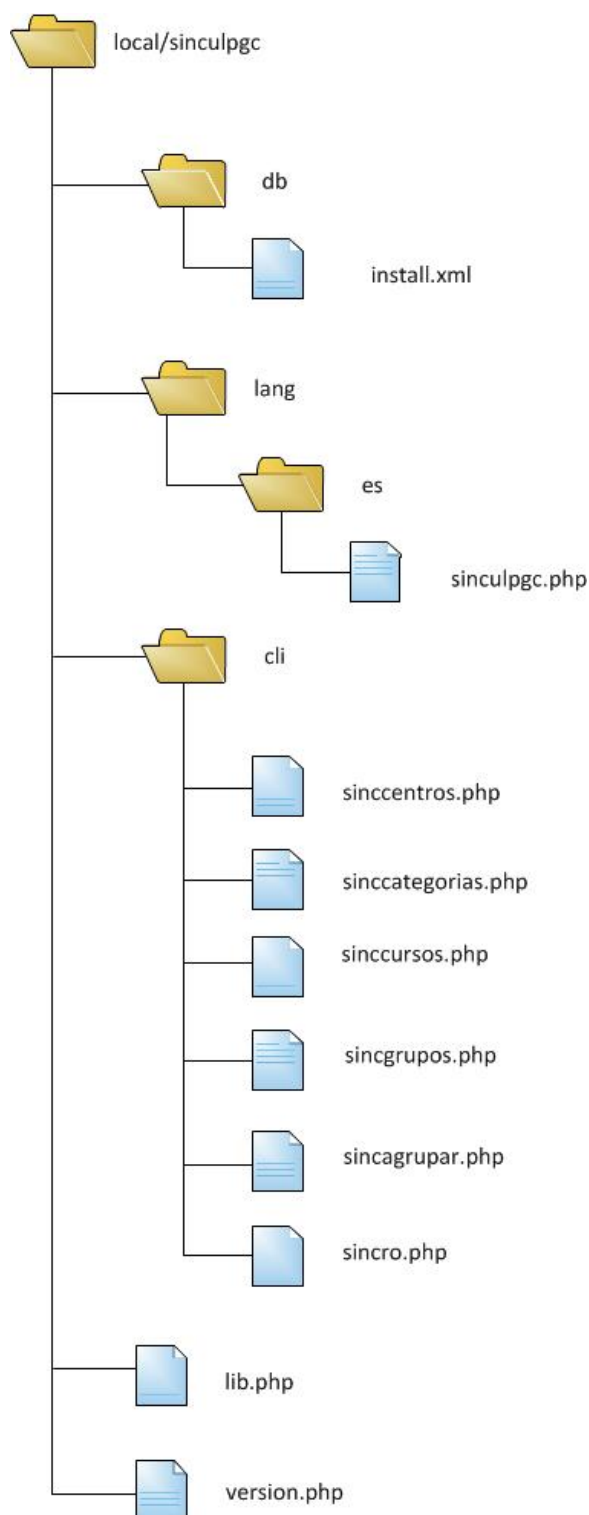


Figura 9-3 Estructura de archivos de extensión local

El fichero *lib.php* comprende las funciones necesarias para conectarse a la base de datos externa.

El fichero *version.php* se describió en **8.2.1 Elementos comunes a todas las extensiones**, es necesario para el gestor de extensiones de Moodle, indicando si se es necesario crear o actualizar alguna tabla en la base de datos.

El archivo *db/install.xml* define la creación de la tabla de centros, que no existe en la instalación estándar de Moodle. Se utiliza para la definición de la tabla **XMLDB**, que es la capa de abstracción de bases de datos de Moodle. En el **Anexo VI Base de datos: guía de estilo y XMLDB** se especifica cómo se define una tabla en XMLDB y consideraciones a tener en cuenta al crear estructuras en la base de datos.

En la archivo *lang/es_utf8/sinculpgc.php* están las cadenas de texto utilizadas en la extensión en idioma español.

Dentro de la carpeta *cli* se ubican los scripts que se ejecutan desde línea de comandos:

- **sincro.php**: ejecuta en lote todos los scripts de sincronización;
- **sinccentros.php**: sincronización de centros;
- **sinccategorias.php**: sincronización de titulaciones (categorías);
- **sincursos.php**: sincronización de asignaturas (cursos);
- **sincgrupos.php**: sincronización de grupos;
- **sincagrupar.php**: sincronización de asignaciones a grupos.

9.4.2 Estructura de los scripts

La estructura de los scripts es muy similar. A continuación se descompone en varios bloques, comunes al resto de scripts, *sinccategorias.php*:

```
// CABECERA

if (! defined ( 'CLI_SCRIPT' )) {

    define ( 'CLI_SCRIPT', true );

}

require_once (dirname ( __FILE__ ) . '/../../../../../config.php');

...

// CONTADORES

$contadores = array (

    'acrear' => 0,

    'creados' => 0,
```

```
        'existe' => 0,
        'avisible' => 0,
        'aeliminar' => 0,
        'eliminados' => 0
    );
    ...
    // GESTIÓN DE REGISTROS
    // Obtención de registros en BBDD externa
    $extdb = db_init();
    $sqlcategoriasulpgc = "SELECT name AS name, faculty_degree as idnumber,
parent, estado
        FROM tmocategorias
        WHERE   plataforma='{ $CFG->plataforma}'   and   aacada='{ $CFG-
>curso_academico}'
        ORDER BY nivel ASC";
    $categoriasulpgc = get_rows_external_db($extdb, $sqlcategoriasulpgc,
'idnumber');
    db_close($extdb);

    // Obtención de registros en Campus Virtual
    $sqlcategoriascv = "SELECT idnumber
        FROM { $CFG->prefix}course_categories";
    $categoriascv = $DB->get_fieldset_sql($sqlcategoriascv);

    // Registros a tratar (Insertar o Eliminar)
    $categoriasadd = array_diff(array_keys(array_filter($categoriasulpgc,
function ($obj)
    {
        if ($obj->estado == 'I')
            return true;
    }
    )
    )
    );
```

```
))), $categoriascv);

$categoriasdel = array_intersect(array_keys(array_filter($categoriasulpgc,
function ($obj)
{
    if ($obj->estado == 'D')
        return true;
})), $categoriascv);

$categoriaskeys = array_merge($categoriasadd, $categoriasdel);

$categorias = array_intersect_key($categoriasulpgc,
array_flip($categoriaskeys));

// TRATAMIENTO DE CADA REGISTRO
foreach ($categorias as $categoria) {

    // Categoría a añadir
    if ($categoria->estado == 'I') {
        ...
    }

    if ($categoria->estado == 'D') {
        ...
    }
}
```

Cabecera

En primer lugar hay que definir la constante *CLI_SCRIPT*, que indica a Moodle que se trata de un script de consola. En caso contrario fallará la ejecución del mismo.

Seguidamente se incluye siempre el fichero *config.php*, que carga las librerías del núcleo y carga las variables globales. A continuación se cargan las librerías necesarias para cada script.

Contadores

Definición de los contadores que se utilizarán para registrar las modificaciones realizadas por el script.

Gestión de registros

En estas líneas se obtienen los registros de la base de datos externa, los registros en el Campus Virtual y se comparan:

- los categorizados para insertar en la base de datos externa y no existen en el Campus Virtual, se insertan en el Campus Virtual.
- los categorizados para eliminar en la base de datos externa y que existen en el Campus Virtual, se eliminan en el Campus Virtual.

Con el fin de optimizar el rendimiento, se realiza una única consulta a la base de datos externa y se utilizan las potentes funciones de arrays de PHP, ganando en velocidad y facilidad de manejo. Se comparan los identificadores únicos de cada registro, que coinciden tanto en la base de datos externa como en Moodle.

Tratamiento de cada registro

Cada registro se trata individualmente, ya sea una nueva inserción o una eliminación. En el caso de que se intente crear una entidad que ya existe o eliminar una que no existe, se pasa al siguiente registro.

Antes de insertar o eliminar cada entidad puede ser necesario realizar algunas comprobaciones y definir algunos valores que no provienen de la base de datos externa. De la misma forma, una inserción o eliminación en ocasiones acarrea operaciones adicionales. En el código fuente se comentan todas estas situaciones.

9.5 EXTENSIÓN DE IDENTIFICACIÓN

Las extensiones de identificación son extensiones de la clase *auth_plugin_base*. La clase, siguiendo la convención de Moodle, se llamará *auth_plugin_casulpgc* y se define en el archivo */auth/casulpgc/auth.php*. A continuación se detallan los detalles de la implementación más relevantes.

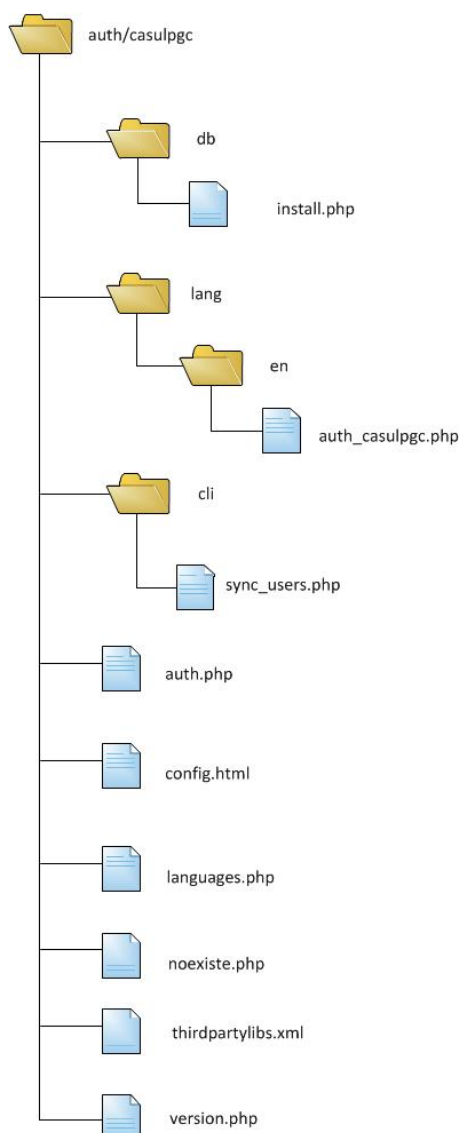


Figura 9-4 Estructura de archivos extensión identificación

9.5.1 Configuración de la extensión

config.html

Página ubicada en */auth/casulpgc/* con las opciones de configuración de la extensión. Esta página se muestra cuando el usuario accede a **Administración del sitio - Extensiones - Identificación – Gestionar identificación – CAS ULPGC (SSO)**.

Contiene un formulario web, mayormente escrito en HTML, donde el administrador introducirá los parámetros de conexión al servidor CAS.

process_config

Es un método de la clase que procesa y almacena los valores de configuración introducidos en la página de configuración de la extensión.

```
...
if (!isset ($config->removeuser )) {
    $config->removeuser = 0;
}
if (!isset($config->logout_return_url)) {
    $config->logout_return_url = '';
}
...
```

9.5.2 Autenticación

El proceso de autenticación, descrito en **8.2.3 Extensión de identificación**, es independiente de la base de datos externa. Verifica que el usuario existe en Moodle y que ha iniciado sesión en el servidor CAS.

loginpage_hook

Este es el método de cada extensión que invoca la página */login/index.php*, donde se puede modificar el comportamiento del proceso de identificación.

Para cumplir con el requisito **F13**, solo se permite el acceso previa autenticación en el servidor CAS, por lo que se chequea tal condición y, si no se cumple, se envía al usuario a la página del Servicio de identificación centralizada de la ULPGC. En caso de que el usuario esté autenticado en el CAS, pero no tenga cuenta en la instancia de Moodle, se le reenvía a una página indicándole esta condición.

```
if (phpCAS::checkAuthentication ()) {
    $frm = new stdClass ();
    $frm->username = phpCAS::getUser ();
    $user = get_complete_user_data ( 'username', $frm->username);
```

```
        if ((! $user) || (($user->auth != 'casulpgc') && ($user->auth !=
'manual')) {
            redirect ( $CFG->wwwroot . '/auth/casulpgc/noexiste.php' );
        }
    }
    return;
```

user_login

Este método devuelve verdadero si el usuario se identifica correctamente. De acuerdo al requisito **F13**, esta autenticación se realiza en el servidor CAS, por lo que se redirige al mismo al usuario no autenticado. Una vez autenticado, el propio servidor CAS devuelve el control a Moodle.

Se apoya en el método *connectCAS*, que crea un cliente y establece la conexión con el servidor CAS.

```
function user_login($username, $password) {
    $this->connectCAS ();

    return phpCAS::isAuthenticated () && (trim ( core_text::strtolower (
phpCAS::getUser () ) ) == $username);
}

function connectCAS() {
...

    if (! $connected) {
        if ($this->config->proxycas) {
            phpCAS::proxy ( $this->config->casversion, $this->config-
>hostname, ( int ) $this->config->port, $this->config->baseuri, false );
        } else {
            phpCAS::client ( $this->config->casversion, $this->config-
>hostname, ( int ) $this->config->port, $this->config->baseuri, false );
        }
        $connected = true;
    }
}
```

```
    }  
...  
}
```

logoutpage_hook

Esta función modifica el comportamiento al finalizar la sesión. En concreto, se cierra la sesión también en el servidor CAS al cerrar la sesión en el Campus Virtual. Se utiliza la API de CAS para PHP, especificando la página a la que se reenvía al usuario tras las desconexión:

```
...  
$this->connectCAS ();  
$redirect = phpCAS::getServerLogoutURL () . '?service=' . urlencode ( $this->config->logout_return_url );  
...  

```

9.5.3 Sincronización con la base de datos externa

Es necesario crear las cuentas de los usuarios de la plataforma, tanto estudiantes como profesores, a partir de la información en la base de datos corporativa. Éste procedimiento se ejecutará de forma automática diariamente, en un horario que no afecte al rendimiento de los sistemas implicados, o manualmente.

sync_users.php

Este es el script que se invoca desde línea de comandos para iniciar la sincronización de usuarios. Al tratarse de un script de línea de comandos, se ubica en la carpeta *auth/casulpgc/cli*.

En última instancia, invoca al método *sync_users*.

sync_users

Gestiona la conexión a la base de datos externa y la creación/eliminación de cuentas.

Tiene una estructura similar a los scripts de la extensión local, pero, por convención, la sincronización de usuarios es un método de la clase de identificación.

La extensión se puede configurar para que elimine las cuentas definitivamente o solamente que las suspenda. Dada esta posibilidad, siempre se verificará, antes de realizar ninguna acción, si la cuenta del usuario ya existe en Moodle.

```
...  
// Verificar existencia de la cuenta del usuario  
$existing_user = $DB->get_record ( 'user', array (   
                                     'username' => strtolower ( $user-  
>username )) );  
...
```

Cuando la cuenta del usuario ya existe, hay dos posibilidades:

- Si no utiliza esta extensión de identificación, se le asigna. De esta forma se controla a los usuarios que provienen de la base de datos externa.
- Si la cuenta está suspendida, se reactiva.

Si la cuenta del usuario no existe, se crea, con la información personal de la base de datos externa y una serie de valores por defecto.

Al eliminar una cuenta solo hay que tener en cuenta la configuración al respecto, eliminándose la cuenta definitivamente o suspendiéndola, según corresponda.

9.6 EXTENSIÓN DE MATRICULACIÓN

Se duplica la carpeta de la extensión *Base de datos externa*, llamándola *sinculpgc*, y, para que Moodle lo reconozca como una nueva extensión de matriculación, modificamos el nombre de la clase en el fichero *enrol/sinculpgc/lib.php*, denominándola *enrol_sinculpgc_plugin*.

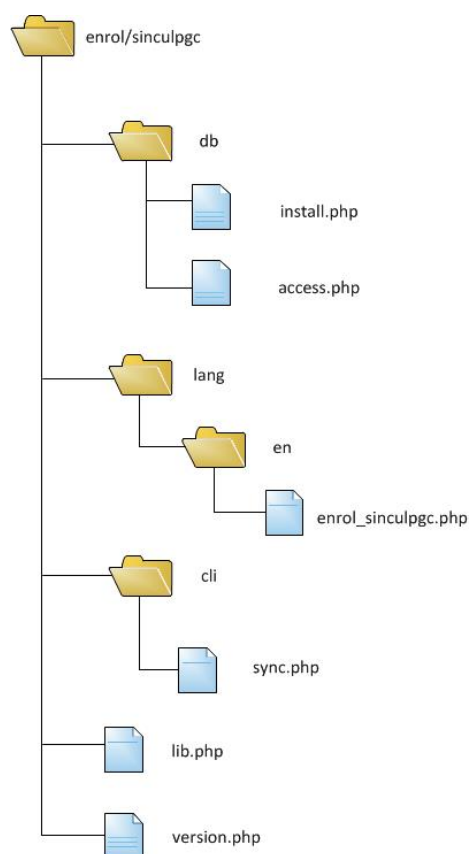


Figura 9-5 Estructura de archivos extensión matriculación

9.6.1 Sincronización con la base de datos externa

Esta es el principal objetivo de la extensión. Al igual que la extensión de identificación, hay un script que se ejecuta desde línea de comandos que invoca el método de la clase que realizar la sincronización.

`sync.php`

Al ejecutarse desde línea de comandos, se ubica en *`enrol/sinculpgc/cli`*.

`sync_enrolment`

Sigue el esquema del resto de scripts. Utilizando arrays se determinan los registros a tratar, para lo cual se emplea la API de enrol de Moodle.

Adicionalmente es necesario definir tres métodos de la clase para facilitar su administración:

- *`instance_deleteable($instance)`*: indica si se permite eliminar este método de matriculación de un curso;

- *allow_unenrol_user(stdClass \$instance, stdClass \$ue)*: para desmatricular usuarios desde la interfaz de la aplicación;
- *get_user_enrolment_actions(course_enrolment_manager \$manager, \$ue)*: devuelve las acciones que puede realizar un usuario sobre una matriculación de esta extensión.

access.php

En este archivo se define un array con las nuevas capacidades inherentes a esta extensión. Las capacidades se asignan a los roles para permitir realizar determinadas acciones a los mismos:

- *enrol/sinculpgc:config*: configuración de la extensión;
- *enrol/sinculpgc:manage*: gestión de las instancias de la extensión;
- *enrol/sinculpgc:enrol*: matricular usuario;
- *enrol/sinculpgc:unenrol*: desmatricular usuario.

9.7 EJECUCIÓN COMPLETA DE LA SINCRONIZACIÓN

Los scripts de sincronización deben ejecutarse secuencialmente, siguiendo el orden establecido en **8.2.5 Secuencia de ejecución**. En la extensión local se ubica el archivo *local/sinculpgc/cli/sincro.php*, responsable de esta tarea. En función de la plataforma en la que se realice la sincronización, determina que entidades (scripts) deben sincronizarse y cuáles no.

Este script debe ejecutarse desde el cron del sistema de uno de los servidores de cada plataforma a sincronizar.

10 EJECUCIÓN DE LAS PRUEBAS

Para facilitar la realización de las pruebas se ha desarrollado un script (*local/sinculpgc/cli/restoremoodlepfc.sh*) que permite reiniciar la plataforma al estado inicial, de tal forma que cada prueba sea independiente de la anterior.

El procedimiento a seguir es el siguiente:

- realizar una instalación limpia de Moodle;
- hacer una copia de la base de datos y del directorio de datos;
- ubicar ambas copias en la misma carpeta que esté el script de restauración;
- realizar la prueba;
- verificar resultado en la plataforma;
- ejecutar el script de restauración.

El script se ha desarrollado a medida para el entorno de la ULPGC, por lo que solo contempla MySQL como SGBD. La ejecución del mismo elimina la base de datos y el directorio de datos, creándolos de nuevo a partir de las copias realizadas tras la instalación de la plataforma.

Como se detalló anteriormente, existe dependencia entre las diferentes entidades que se sincronizan. Por esta razón las pruebas se realizaron de forma incremental, añadiendo paulatinamente la sincronización de nuevos elementos. De esta forma, la secuencia de ejecución de las tandas de pruebas fue la siguiente:

- sincronización de centros;
- sincronización de centros y titulaciones;
- sincronización de centros, titulaciones y asignaturas;
- sincronización de centros, titulaciones, asignaturas y grupos;
- sincronización de centros, titulaciones, asignaturas, grupos y usuarios;
- sincronización de centros, titulaciones, asignaturas, grupos, usuarios y matrículas;

- sincronización de centros, titulaciones, asignaturas, grupos, usuarios, matrículas y asignaciones de grupos.

10.1 REALIZACIÓN DE LAS PRUEBAS

La sincronización de la plataforma depende de la información en la base de datos externa, por lo que es en ésta donde se prepara cada escenario, introduciendo los registros a sincronizar.

En lugar de insertar solo los registros necesarios para realizar las pruebas, se utilizó una carga real, verificando que se daban todas las situaciones descritas en **8.3.1 Pruebas Unitarias**.

A continuación se muestran las fichas de las pruebas propuestas en **7.4 Especificación del Plan de Pruebas** con los resultados obtenidos.

10.1.1 Centros

Los centros no se muestran en la interfaz de usuario. Los usan otros módulos con fines diversos. Para verificar que la realización de la prueba ha sido correcta es necesario consultar directamente la tabla en la base de datos.

<i>Escenario E1: Gestionar centros</i>	
<i>Caso de Prueba: CPI.1</i>	
Entrada	Resultado Obtenido
Añadir un centro no existente	Registro de centro creado correctamente.
<i>Caso de Prueba: CPI.2</i>	
Entrada	Resultado Obtenido
Añadir centro que ya existe	No se realiza ninguna modificación.
<i>Caso de Prueba: CPI.3</i>	
Entrada	Resultado Obtenido
Eliminar centro	No se encuentra el registro en la tabla, habiéndose verificado previamente su existencia.

<u>Caso de Prueba: CP1.4</u>	
Entrada	Resultado Obtenido
Eliminar centro no existe	No se realiza ninguna modificación en la tabla.

10.1.2 Titulaciones

Las titulaciones se crean como categorías en el Campus Virtual. Para ver la configuración se accede a la opción *Administración del sitio -> Cursos -> Administrar cursos y categorías*.

Recordar que con el fin de no perder información las categorías no se eliminan, se ocultan.

<u>Escenario E2: Gestionar titulaciones</u>	
<u>Caso de Prueba: CP2.1</u>	
Entrada	Resultado Obtenido
Añadir una titulación no existente	Verificado que se crea la categoría dentro de la categoría de centro correspondiente.
<u>Caso de Prueba: CP2.2</u>	
Entrada	Resultado Obtenido
Añadir una titulación que ya existe	No se duplica la categoría ni se altera la ya existente.
<u>Caso de Prueba: CP2.3</u>	
Entrada	Resultado Obtenido
Eliminar una titulación	La categoría, previamente verificada su visibilidad, está oculta, así como todos sus cursos.
<u>Caso de Prueba: CP2.4</u>	
Entrada	Resultado Obtenido

Eliminar titulación no existe	No hay modificación alguna en el Campus Virtual.
-------------------------------	--

10.1.3 Asignaturas

Las asignaturas se crean como cursos pertenecientes a la categoría que representa la titulación en la que se imparten. Tras lanzar una prueba accedemos a **Administración del sitio - Cursos - Administrar cursos y categorías** para comprobar la existencia y visibilidad de los distintos cursos.

Recordar que con el fin de no perder información las cursos no se eliminan, se ocultan.

<u>Escenario E3: Gestionar asignaturas</u>	
<u>Caso de Prueba: CP3.1</u>	
Entrada	Resultado Obtenido
Añadir asignatura no existente	Se creó un curso con el nombre de la asignatura, dentro de la categoría que representa la titulación que la oferta.
<u>Caso de Prueba: CP3.2</u>	
Entrada	Resultado Obtenido
Añadir asignatura que ya existe	Ni se duplicó el curso ni se modificó el existente.
<u>Caso de Prueba: CP3.3</u>	
Entrada	Resultado Obtenido
Eliminar asignatura	El curso, previa verificación de su visibilidad, está oculto.
<u>Caso de Prueba: CP3.4</u>	
Entrada	Resultado Obtenido
Eliminar asignatura no existe	No hay modificación alguna en el Campus Virtual.

10.1.4 Grupos

Los grupos se crean dentro de los cursos. Los grupos de un curso se pueden visualizar, una vez dentro del curso, en **Administración del curso - Usuarios - Grupos**. Los participantes de un grupo se muestran en la misma pantalla, seleccionando un grupo y pulsando el botón *Mostrar miembros del grupo*. También se puede ver en qué grupos está asignado un usuario mostrando su perfil desde el curso, desde el menú **Navegación - Curso actual - Participantes** pulsando sobre un usuario.

Es necesario consultar el campo *component* de la tabla *groups* para conocer el origen de la creación del grupo. Cuando se crean durante el proceso de sincronización tiene el valor *sinculpgc*.

Al realizar las pruebas con una cantidad de registros similar a la real se detectó que el script de sincronización de asignaciones abortaba debido a un fallo de memoria. El número de registros de la base de datos externa era demasiado grande para gestionarlo con una sola llamada. Esta es la razón por la que la sincronización de asignaciones se realiza por categorías, para reducir el número de registros a obtener desde la base de datos externa.

<u>Escenario E4: Gestionar grupos</u>	
<u>Caso de Prueba: CP4.1</u>	
Entrada	Resultado Obtenido
Añadir un grupo no existente	Se creó un grupo dentro del curso correspondiente a la asignatura a la que pertenece el grupo.
<u>Caso de Prueba: CP4.2</u>	
Entrada	Resultado Obtenido
Añadir grupo que ya existe	No se creó ningún grupo y no se modificó el que ya existía.
<u>Caso de Prueba: CP4.3</u>	
Entrada	Resultado Obtenido
Eliminar grupo	El grupo, previa verificación de existencia, se eliminó.

<u>Caso de Prueba: CP4.4</u>	
Entrada	Resultado Obtenido
Eliminar grupo no existe	No se apreció ningún cambio.
<u>Caso de Prueba: CP4.5</u>	
Entrada	Resultado Obtenido
Añadir grupo académico que se creó manualmente	Se comprueba que un grupo creado manualmente, después de la sincronización, pasa a ser un grupo sincronizado.
<u>Caso de Prueba: CP4.6</u>	
Entrada	Resultado Obtenido
Asignar usuario matriculado en asignatura a grupo	El usuario se asignó al grupo correspondiente.
<u>Caso de Prueba: CP4.7</u>	
Entrada	Resultado Obtenido
Desasignar usuario matriculado en asignatura a grupo	El usuario no tiene asignado el grupo.
<u>Caso de Prueba: CP4.8</u>	
Entrada	Resultado Obtenido
Asignar usuario matriculado en asignatura a grupo no existe	No se realizó ninguna acción en el Campus Virtual.

10.1.5 Usuarios

Estas pruebas engloban la creación de cuentas de usuario, su acceso y la matriculación en los cursos que corresponda.

Los usuarios no se borran del Campus Virtual para evitar pérdidas de información. En su lugar se les suspende la cuenta, con la consiguiente pérdida de acceso al Campus Virtual.

<u>Escenario E5: Gestionar usuarios</u>	
<u>Caso de Prueba: CP5.1</u>	
Entrada	Resultado Obtenido
Añadir usuario que no existe	En el Campus Virtual se creó la cuenta del usuario
<u>Caso de Prueba: CP5.2</u>	
Entrada	Resultado Obtenido
Añadir usuario que existe	No se realizó ninguna modificación en el Campus Virtual
<u>Caso de Prueba: CP5.3</u>	
Entrada	Resultado Obtenido
Eliminar usuario que existe	Se suspendió la cuenta del usuario.
<u>Caso de Prueba: CP5.4</u>	
Entrada	Resultado Obtenido
Verificar acceso correcto	El usuario autenticado en CAS pudo acceder al Campus Virtual.
<u>Caso de Prueba: CP5.5</u>	
Entrada	Resultado Obtenido

Ejecución de las pruebas

Verificar restricción acceso	El usuario, sin estar autenticado en CAS, fue redirigido a la página de identificación. Tras autenticarse se dirige al Campus Virtual.
<i><u>Caso de Prueba: CP5.6</u></i>	
Entrada	Resultado Obtenido
Matricular usuario en curso	El usuario está matriculado en el curso y pudo acceder.
<i><u>Caso de Prueba: CP5.7</u></i>	
Entrada	Resultado Obtenido
Desmatricular usuario de curso	El usuario dejó de estar matriculado y no puede acceder al curso.

11 CONCLUSIONES Y LÍNEAS DE TRABAJO

FUTURAS

11.1 CONCLUSIONES

Este proyecto se realizó como parte de un proyecto de mayor tamaño que abarcaba la puesta en marcha del Campus Virtual de la ULPGC 2014/15.

A mediados de julio del 2014 ya estaban instaladas las nuevas plataformas, y se utilizó este proceso de sincronización para cargar la información académica en las mismas. En ese momento se cargaron los centros, titulaciones, asignaturas, grupos y se dieron de alta y asignaron a sus titulaciones y grupos los profesores. En septiembre, poco antes del inicio de las clases, se dieron de alta y asignaron a titulaciones y grupos también los alumnos.

Desde entonces está en funcionamiento el proceso de sincronización, ejecutándose a diario en las plataformas siguientes:

- Grado y Posgrado
- Teleformación
- Otras enseñanzas
- Social

El número de incidencias relacionadas con la sincronización se han reducido en un 100% con respecto al curso académico 2013/14. Todas las incidencias hasta el momento han tenido causa académica o en las especificaciones de cómo determinar la información a sincronizar.

Además, se han solventado otros problemas recurrentes en cursos anteriores:

- notable reducción en el tiempo de ejecución, facilitando su ejecución manual en caso de querer sincronizar una plataforma antes de la hora programada;
- un fallo en la sincronización (caída de base de datos local, error en acceso a base de datos externa, etc.) no implica trabajo adicional. Basta con relanzar el proceso.

Un beneficio adicional es el ahorro de recursos en la resolución de incidencias. Los problemas de sincronización en cursos anteriores demandan la atención del personal y la consiguiente inversión de tiempo, que ralentizaba o impedía la realización de otros proyectos.

La validez de la información académica en las plataformas del Campus Virtual incluso está creando la tendencia entre los usuarios de utilizarlo como referencia en lugar de otras herramientas disponibles: listados de alumnos, calificaciones, pertenencia a grupos, etc. A su vez, esta misma tendencia obliga a garantizar que el proceso de sincronización se mantiene a este nivel y, en la medida de lo posible, mejore.

11.2 LÍNEAS DE TRABAJO FUTURAS

Actualmente se sigue mejorando el proceso de sincronización, añadiendo más información o nuevas características. Las principales líneas de trabajo son las siguientes:

- **Nueva información:** se añade información a las entidades que ya se sincronizan, como por ejemplo identificar alumnos morosos o añadir carga docente de los profesores, y nuevas entidades, como los departamentos.
- **Sincronización desde interfaz de usuario:** de esta forma los administradores de cada plataforma podrían lanzar el proceso de sincronización manualmente, sin requerir la intervención de los administradores de sistemas.
- **Opciones de configuración:** las opciones de configuración en esta versión son bastante limitadas. Los valores por defecto de las entidades que se sincronizan están definidos en el código fuente, mientras que si existiese una interfaz en la plataforma, los administradores tendrían la posibilidad de modificarlos.
- **Granular el proceso de sincronización:** apoyándose en dos de las líneas anteriores, disponer de una interfaz y más opciones de configuración, sería posible sincronizar solo aquellas entidades que se precisen en un momento determinado o seleccionar subconjuntos de información de las mismas. Por ejemplo, sincronizar solo la asignación de grupos de una asignatura o dar de alta a los profesores antes del inicio del curso.
- **Actualización de registros:** en esta versión del proceso de sincronización no se actualizan las entidades ya creadas en el Campus Virtual. Si un usuario cambia de nombre o una titulación de cuatrimestre no se refleja en el Campus Virtual. Se ha hecho

ya alguna aproximación utilizando Servicios Web con buenos resultados, pero habría que extenderlo a toda la información susceptible de actualización.

- **Acceso al log de sincronización desde interfaz web:** el log del proceso de sincronización no fue una solicitud de los usuarios, por lo que se implementó su uso únicamente con el propósito de facilitar la depuración de posibles errores por parte de los administradores de sistemas o los desarrolladores, quienes deben acceder directamente a la tabla de la base de datos para obtener esta información. En ocasiones los administradores de la plataforma o incluso usuarios con otros roles, profesores y estudiantes, han solicitado información relativa a la sincronización, peticiones que justifican que se facilite el acceso a esta información por sí mismos empleando el mismo mecanismo que emplean para acceder a otras áreas del log de Moodle, denominado *Informes o Registros* en el bloque *Administración*.
- **Sincronización en tiempo real:** este sería el proyecto más ambicioso y dejaría el proceso de sincronización en un segundo plano o incluso lo relevaría. La idea base es modificar las aplicaciones de gestión tradicionales para que cada vez que se añada información académica ésta se añada en la plataforma del Campus Virtual que corresponda. Moodle dispone de una capa de Servicios Web y una API para desarrollar cuántos sean precisos que facilitarían esta labor. Ejemplos:
 - cuando un alumno formaliza la matrícula en la administración, automáticamente se matricula en los cursos correspondientes en el Campus Virtual;
 - si en Ordenación Académica se asigna docencia a un profesor, al finalizar el trámite se traspasa esta información al Campus Virtual.

12 BIBLIOGRAFÍA

- Büchner, A. (2011). *Moodle Administration*. Packt Publishing.
- Clarenc, C. A. (2013). *Instrumento de evaluación y selección de sistemas de gestión de aprendizaje y otros materiales digitales*. Congreso Virtual Mundial de e-Learning 2013.
- Hunt, T. (2012). *The Architecture of Open Source Applications Volume II*. (A. Brown, & G. Wilson, Edits.)
- Moodle. (15 de Agosto de 2014). *Data manipulation API*. Recuperado el 2014, de Moodle: https://docs.moodle.org/dev/Data_manipulation_API
- Moodle. (12 de 08 de 2014). *Developer documentation*. Obtenido de Moodle: http://docs.moodle.org/dev/Developer_documentation
- Moodle. (19 de Junio de 2014). *PHPUnit*. Recuperado el 2014, de Moodle: <http://docs.moodle.org/dev/PHPUnit>
- Moodle. (26 de Diciembre de 2014). *Seguridad*. Recuperado el 2014, de Moodle: <http://docs.moodle.org/dev/Security>
- Moodle. (s.f.). *Moodle architecture*. Recuperado el 17 de 12 de 2014, de http://docs.moodle.org/dev/Moodle_architecture#Moodle_as_a_modular_system

ANEXO I: INSTALACIÓN DE MOODLE

La complejidad de la instalación de Moodle varía notablemente en función de la infraestructura implicada y la personalización de la plataforma.

Para este proyecto se cuenta con la colaboración del Servicio de Informática de la ULPGC, que facilita la instalación y configuración del entorno LAMP (Linux, Apache, MySQL y PHP) necesario para la ejecución de Moodle. Disponiendo de un entorno que cumple con los requisitos a nivel de servidores, podemos realizar una instalación rápida de Moodle siguiendo las instrucciones https://docs.moodle.org/27/en/Installation_Quickstart. En el anexo XX se incluyen las características del entorno así como los requisitos mínimos de Moodle.

En esta instalación no se considerarán aspectos de seguridad o rendimiento, sino disponer de una instalación funcional que permita desarrollar las extensiones necesarias.

CÓDIGO DE MOODLE

Hay varias alternativas para disponer del código de Moodle. En el servidor donde es va a instalar disponemos de un cliente Git, por lo que se puede obtener desde el repositorio Git oficial:

```
git clone -b MOODLE_27_STABLE git://git.moodle.org/moodle.git
```

Con esta instrucción desde línea de comandos se crea la carpeta *moodle* y se obtiene una copia completa del repositorio de Moodle y se cambia a la rama estable 2., versión actual en el momento de realizar este proyecto.

Verificar que el servidor web no tiene permiso para escribir en ningún fichero de Moodle.

CREAR UNA BASE DE DATOS

Se dispone de las credenciales de administrador en la base de datos MySQL destinada al proyecto.

Para crear una base de datos:

```
mysql> CREATE DATABASE basedatos DEFAULT CHARACTER SET UTF8 COLLATE
utf8_unicode_ci;
```

Se crea un usuario y se le asignan todos los permisos sobre la base de datos recién creada:

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY
TABLES,DROP,INDEX,ALTER ON basedatos.* TO moodleuser@localhost IDENTIFIED BY
'yourpassword';
```

CREAR DIRECTORIO DE DATOS

Es necesario un directorio vacío para almacenar los ficheros de Moodle. Por motivos de seguridad, no debe estar en el área pública del servidor web, pero si debe tener permisos para que el usuario del servidor web pueda escribir en él. Basta con que el usuario del servidor web sea el propietario o dar permiso de escritura a todos en el directorio.

Se opta por crear el directorio en la raíz de la máquina donde se ubica el servidor web y asignar el grupo *apache*, al que pertenece el usuario del servidor web, al directorio.

```
mkdir /directoriodatos
chown usuario:apache /directoriodatos
chmod 775 /directoriodatos
```

INSTALACIÓN

Se puede instalar desde el navegador o desde línea de comandos. En este caso, hay que ejecutar el script `install.php` y seguir las instrucciones que se muestran en consola:

```
php /path/to/moodle/admin/cli/install.php
```

Este script crea el fichero de configuración *config.php*.

Durante el proceso de instalación se habrán generado credenciales para un usuario con permisos de administración, por defecto *admin*. Con esas credenciales se puede acceder desde el navegador.

ANEXO II: SUBVERSION

Cada usuario dispone de un directorio de trabajo propio en el entorno de desarrollo. Se crea una instalación de Moodle y un usuario para cada desarrollador en la máquina de desarrollo.

CREACIÓN DEL DIRECTORIO DE TRABAJO

Para crear un nuevo directorio de trabajo, es decir, una copia del código fuente, en */var/www/html* lanzamos el comando

```
# svn checkout file:///var/svn/repos/moodle/trunk moodle_user
```

donde *user* será el usuario con el que se accede al sistema el desarrollador; por ejemplo, **moodle_vdeniz**.

FLUJO DE TRABAJO GENERAL

Actualizar la copia de trabajo

Para actualizar la copia de trabajo con la última versión de los ficheros del proyecto, con las posibles modificaciones que hayan realizado otros miembros del equipo de desarrollo, hay que ejecutar el siguiente comando desde el directorio de trabajo:

```
# svn update
```

Hacer cambios en la copia de trabajo

Modificar el contenido de ficheros no requiere ninguna acción especial. Los comandos de Subversion solo tienen efecto sobre el repositorio del mismo, no copian, mueven ni borran ficheros de la copia de trabajo.

Es preciso señalar que las modificaciones no se efectúan en el sistema de control de versiones hasta que se efectúa un *commit*.

Añadir o eliminar ficheros

Si se añade o elimina un fichero o directorio a la copia de trabajo (físicamente), se notifica al sistema de control de versiones con:


```
# svn add foo
```

```
# svn delete foo
```

El fichero o directorio *foo* se eliminará o añadirá al control de versiones en el próximo *commit*.

Copiar o mover ficheros

Es posible copiar o mover ficheros junto con su historial de modificaciones, sin necesidad de hacerlo físicamente, con los comandos:

```
# svn copy foo bar
```

```
# svn move foo bar
```

Examinar los cambios

Para examinar los cambios que se van a efectuar en el repositorio antes de hacerlos efectivos:

```
# svn status
```

Es posible ver las diferencias existentes entre los ficheros del repositorio y los modificados en la copia local:

```
# svn diff
```

Deshacer cambios

Es posible cancelar las modificaciones realizadas en los ficheros o directorios de trabajo y volver al estado anterior:

```
# svn revert foo
```

Confirmar los cambios

Una vez realizado los cambios necesarios, hay que hacerlos efectivos en el repositorio, añadiendo algún mensaje significativo:

```
# svn commit -message "Añadida la actualización automática de los departamentos"
```

Resolver conflictos

Cuando intentamos actualizar en el repositorio un fichero que ya ha modificado otro desarrollador, se producen conflictos que, en ocasiones, tenemos que resolver manualmente, bien descartando algunas modificaciones o combinando dos ficheros. Consultar el manual para resolver estas situaciones.

ANEXO III: INSTALACIÓN Y CONFIGURACIÓN DE ECLIPSE

En el momento de desarrollo de este proyecto, la versión disponible es Eclipse Juno (4.2), que se puede descargar desde la web del proyecto (Eclipse es algo más que un IDE), <http://www.eclipse.org/downloads/>. Para su instalación se pueden seguir las instrucciones en <http://wiki.eclipse.org/Eclipse/Installation>.

INSTALAR MÁQUINA VIRTUAL DE JAVA (JVM, JAVA VIRTUAL MACHINE)

Eclipse, al ser una aplicación Java, requiere la instalación de un entorno de ejecución Java (JRE, Java runtime environment). Independientemente del sistema operativo en el que se vaya a utilizar, es necesario instalar una JVM. Se puede instalar un JRE o un JDK (Java Development Kit), pero éste último solo es necesario si se va a programar en Java.

La versión disponible de Java es la Version 7 Update 21, que se puede descargar desde <http://www.java.com/es/download/>. Una vez descargada, la instalación es trivial, necesitando únicamente especificar el directorio de instalación.

INSTALAR ECLIPSE

Desde el enlace mencionado en la introducción de este epígrafe se descarga un fichero comprimido. Este fichero se descomprime en un directorio de libre elección, donde se alojará el archivo ejecutable de la aplicación.

Añadir plugins

Para mejorar la funcionalidad que ofrece Eclipse por defecto, adaptándola a las necesidades de este proyecto, añadimos los siguientes plugins:

- PDT: añade a Eclipse las características básicas de edición de código para desarrollar en PHP:
 - Edición de código PHP con resaltado de sintaxis y coloreado de código
 - Completado de código PHP, plantillas y autoformato

- Depurado local y remoto con soporte para XDebug y Zend Debugger
- Web tools: añade herramientas de desarrollo web. Incluye fuentes y editores para diferentes lenguajes, asistentes y aplicaciones preinstaladas para facilitar el desarrollo.
- Subversive: habilita el uso de Subversion, sistema de control de versiones centralizado que utiliza la UPLGC para gestionar el código fuente de Moodle, desde la interfaz de Eclipse.

Los plugins en Eclipse se instalan desde la opción del menú **Help->Install New Software...**

Configuración

A las opciones de configuración de Eclipse se accede desde **Window->Preferences**. Hay múltiples opciones agrupadas por categorías que se muestran en la parte izquierda en forma de árbol. A continuación se especifican aquellas preferencias personales especificadas.

General->Appearance->Colors and Fonts

En esta opción se pueden modificar la fuente, su tamaño y color, bajo la categoría *Basic*, del que heredan el resto, seleccionar *Text Font*.

General->Content Types

Seleccionar *PHP Source File* y añadir **.html* en la parte inferior.

General->Editors->File Associations

Añadir en la parte superior **.php*. En la parte inferior seleccionar *PHP Editor* y escogerlo por defecto. Seleccionar en la parte superior **.html* y añadir *PHP Editor* como editor por defecto. Facilita la edición de los ficheros HTML de Moodle, que generalmente contienen código PHP.

General -> Editors -> Text Editors

Marcar *Show line numbers* para ver los números de línea en el margen izquierdo.

PHP -> Code style -> Formatter

Establecer los tabuladores como espacios y especificar el número de espacios de indentación. El código de Moodle usa 4 espacios.

PHP -> Editor -> Save Actions

Marcar *Remove trailing whitespace*.

PHP -> Editor -> Templates

En esta opción se pueden definir plantillas de código. Hay una función muy útil en Moodle que muestra el contenido de una variable, que se usa con bastante frecuencia, para la que vamos a definir una plantilla.

Name	po
Description	Muestra información estructurada sobre una variable
Pattern	<code>print_object(\${ word_selection }\${ cursor }); // DONOTCOMMIT</code>

ANEXO IV: GUÍA DE ESTILO DE CÓDIGO

Este apéndice es un extracto de la guía http://docs.moodle.org/dev/Coding_style, con las directrices relevantes y de interés para la realización de este proyecto.

HERRAMIENTAS

Hay dos herramientas disponibles útiles para escribir código utilizando esta guía. Estas herramientas realizan distintas comprobaciones, recomendándose utilizar ambas:

- **Code checker:** http://moodle.org/plugins/view.php?plugin=local_codechecker
- **Marina Glancy's Moodle PHPdoc checker:** https://github.com/marinaglancy/moodle-local_moodlecheck

FORMATO DE ARCHIVO

Etiquetas PHP

Utilizar siempre etiquetas largas. Sin embargo, para evitar problemas con los espacios en blanco, no incluir la etiqueta de cierre al final del archivo.

```
<?php  
  
require('config.php');
```

Justificación

Utilizar una separación de 4 espacios sin tabulador. Los editores se deben configurar para tratar los tabuladores como espacios en blanco para prevenir la introducción de tabuladores en el código fuente. No justificar el nivel principal del script:

```
<?php  
require('config.php');  
$a = required_param('a', PARAM_INT);  
if ($a > 10) {  
    call_some_error($a);  
} else {
```

```
do_something_with($a);  
}
```

Ajuste de línea

Se recomiendan 132 caracteres, no superando los 180.

Cuando se divide una línea en dos, la segunda se justifica con 8 espacios en lugar de 4, resultando más sencillo apreciar la diferencia entre líneas divididas y bloques justificados.

```
if (a_long_condition() &&  
    a_nother_long_condition()) {  
    do_something();  
}
```

Es una buena práctica evitar dividir líneas en estructuras de control.

Arrays

Los arrays asociativos son una excepción a la regla de los 8 espacios. La disposición correcta sería:

```
$plugininfo['preferences'][$plugin] = array(  
    'id' => $plugin,  
    'link' => $pref_url,  
    'string' => $modulenamestr  
);
```

Alinear el símbolo “=>” es opcional. Los arrays pequeños se pueden declarar en una sola línea.

Declaración de funciones

Si una función tiene muchos parámetros, se justifican todos en línea con el primer parámetro:

```
public function graded_users_iterator($course, $grade_items = null,
```

```
$sortfield1 = 'lastname',  
$sortorder1 = 'ASC',  
$sortfield2 = 'firstname',  
$sortorder2 = 'ASC') {
```

Estructuras de control

Si existen muchas condiciones en una estructura de control, se sugiere asignar valores a algunas variables antes de la estructura para mejorar la legibilidad:

```
$coursecategory = $element['object']->is_course_item() or  
$element['object']->is_category_item();  
$scalevalue = in_array($element['object']->gradetype,  
array(GRADE_TYPE_SCALE, GRADE_TYPE_VALUE));  
  
if ($coursecategory and $scalevalue) {
```

Terminación de línea

Se usa el formato de texto estándar de Unix. Las líneas deben terminar solo con un salto de línea (linefeed, LF). Los saltos de línea se representan en ordinal como 10 y en hexadecimal como 0x0A.

No usar restornos de carro (carriage returns, CR), como en Macintosh, ni la combinación CRLF, habitual en Windows.

Las líneas no deberían contener espacios finales, pudiéndose configurar el editor para eliminar los espacios finales en el momento de guardar el archivo.

CONVENCIONES DE NOMBRES

Archivos

Los archivos deberían:

- Ser palabras completas en inglés

- Ser lo más corto posibles
- Usar solo letras en minúsculas
- Usar las extensiones .php, .html, .js, .css o .xml

Clases

Siempre deberían ser palabras minúsculas en inglés, separadas por subrayados:

```
class some_custom_class {
    function class_method() {
        echo 'foo';
    }
}
```

Cuando se creen nuevas instancias usar siempre (), aunque el constructor no requiera parámetros:

```
$instance = new some_custom_class();
```

Si se desea utilizar un objeto plano de ninguna clase, por ejemplo al preparar datos para insertar en la base de datos utilizando el método `$DB->insert_record()`, usar la clase PHP estándar **stdClass**. Por ejemplo:

```
$row = new stdClass();
$row->id = $id;
$row->field = 'something';
$DB->insert_record('table', $row);
```

Funciones y métodos

Los nombres de las funciones serán palabras minúsculas en inglés, empezando con el nombre del módulo para evitar conflictos entre módulos. Las palabras se separarán con subrayados. Los nombres de funciones deberían facilitar la comprensión de su utilidad.

No dejar espacios entre el nombre de 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 parámetros serán siempre palabras minúsculas en inglés (en ocasiones más de una, como en *\$initialvalue*), con valores por defecto siempre que sea posible. Usar *null* en lugar de *false* para aquellos casos en los que no se requiera un valor por defecto. Sin embargo, si un parámetro es de tipo booleano, y su valor por defecto es *true* or *false*, se considera válido utilizar uno de estos valores.

```

public function foo($required, $optional = null)

```

Variables

Los nombres de variables debería ser fáciles de leer, significativas palabras minúsculas en inglés. Si realmente es necesario utilizar más de una palabra, se unirán, manteniendo el nombre tan corto como sea posible. Usar plurales para los arrays de objetos.

```

GOOD: $quiz
GOOD: $errorstring
GOOD: $assignments (for an array of objects)
GOOD: $i (but only in little loops)

```

Las variables principales de Moodle se identifican usando mayúsculas (*\$CFG*, *\$SESSION*, *\$USER*, *\$COURSE*, *\$SITE*, *\$PAGE*, *\$DB* y *\$THEME*). No crear ninguna más.

Constantes

Siempre en mayúsculas, empezando con el nombre del módulo. Utilizar palabras separadas con subrayados.

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

Booleanos

true, **false** y **null** siempre en minúsculas.

Cadenas

El rendimiento en el uso de cadenas no es un problema en PHP, por lo que los siguientes criterios persiguen únicamente mejorar la legibilidad del código.

Comillas simples

Utilizar siempre comillas simples cuando una cadena es literal, o contiene varias comillas dobles (HTML):

```
$a = 'Example string';
echo '<span class="'.s($class).'"></span>';
$html = '<a href="http://something" title="something">Link</a>';
```

Comillas dobles

Utilizar cuando sea necesario incluir variables sencillas o varias comillas simples.

```
echo "<span>${string}</span>";
$statement = "You aren't serious!";
```

Consultas SQL complejas siempre utilizarán comillas dobles.

```
$sql = "SELECT e.*, ue.userid
FROM {user_enrolments} ue
JOIN {enrol} e ON (e.id = ue.enrolid AND e.enrol = 'self')
JOIN {user} u ON u.id = ue.userid
WHERE :now - u.lastaccess > e.customint2";
```

Sustitución de variables

Pueden utilizarse cualquiera de las alternativas siguientes:

```
$greeting = "Hello $name, welcome back!";  
  
$greeting = "Hello {$name}, welcome back!";
```

Concatenación de variables

Las cadenas se concatenan con el operador “.”.

```
$longstring = $several.$short.'strings';
```

Si las líneas son demasiado largas, dividir la sentencia en varias líneas para mejorar la legibilidad.

En este caso poner “.” Al final de cada línea.

```
$string = 'This is a very long and stupid '.$editorname.  
         " couldn't think of a better example at the time.";
```

El operador “.” puede utilizarse sin espacios, como en los ejemplos anteriores, o con espacios antes y después, a elección del desarrollador.

Cadenas de idioma

Mayúsculas

Las cadenas de idioma “Siempre tendrán este formato” y “Nunca Utilizarán Este Otro”. Las letras mayúsculas solo deberán usarse al iniciar una frase o nombre propio.

Estructura

Las cadenas no deben definirse con el objetivo de concatenarlas en la interfaz de usuario, puesto que podría causar problemas en diferentes idiomas. Cada cadena debe ser completa por sí misma.

ARRAYS

Arrays indexados numéricamente

No se admiten números negativos como índices. Un array indexado puede comenzar con cualquier número positivo; sin embargo, no se recomienda el uso de índices base superiores a 0. Cuando se declara un array indexado con la función *array*, se debe dejar un espacio después de cada coma para facilitar la legibilidad.

```
$myarray = array(1, 2, 3, 'Stuff', 'Here');
```

Son válidos los arrays definidos en varias líneas, pero cada línea a partir de la primera se justificará con 8 espacios:

```
$myarray           =           array(
    1,              2,              3,              'Stuff',      'Here',
    $a, $b, $c, 56.44, $d, 500);
```

Arrays asociativos

Usar múltiples líneas si con ello se facilita su comprensión:

```
$myarray           =           array(
    'firstkey'      =>           'firstvalue',
    'secondkey'     =>           'secondvalue'
);
```

CLASES

Declaración

- Los nombres seguirán las convenciones de nombres de Moodle.
- La llave de inicio se escribirá siempre junto al nombre de la clase.
- Cada clase debe tener un bloque de documentación conforme al estándar PHPDocumentor.
- Todo el código de una clase debe estar justificado con cuatro espacios.
- Aunque está permitido, se desaconseja añadir código adicional a la clase en el mismo archivo. Si se da el caso, separar la clase del código adicional utilizando dos líneas en blanco.

```

/**
 *      Documentation      Block      Here
 */
class      sample_class      {
    //      All      contents      of      class
    //      must      be      indented      4      spaces.
}

```

Propiedades

- Los nombres de las propiedades seguirán las convenciones de nombres de Moodle.
- Cualquier propiedad declarada en una clase debe incluirse al inicio de la misma, antes de la declaración de cualquier método.
- No se permite usar el constructor *var*. La visibilidad de las propiedades se declarará usando los modificadores **private**, **protected** o **public**. Se permite habilitar el acceso directo a propiedades declarándolas como **public**, pero se desaconseja en favor del uso de métodos de acceso (*set/get*).

FUNCIONES Y MÉTODOS

Declaración

- Los nombres seguirán las convenciones de nombres de Moodle.
- Los métodos dentro de las clases siempre deben declarar su visibilidad usando los modificadores **public**, **private** o **protected**.
- La llave de inicio se escribirá siempre en la misma línea que el nombre de la función.
- No dejar espacios entre el nombre de función y los paréntesis para los argumentos.
- El valor de retorno no irá entre paréntesis. Debe devolverse solo un tipo de dato, desaconsejándose devolver diferentes tipos.

```

/**
 * Documentation Block Here
 */
class sample_class {

```

```

/**
 * Documentation Block Here
 */
public function sample_function() {
    // All contents of function
    // must be indented four spaces.
    return true;
}
}

```

Uso

Los argumentos pasados a una función se separarán utilizando un espacio tras la coma.

```
three_arguments(1, 2, 3);
```

ESTRUCTURAS DE CONTROL

If / else

- Usar un espacio antes y después de la expresión de control entre paréntesis, separando los operadores con espacios dentro de los paréntesis.
- Utilizar paréntesis interiores para mejorar el agrupamiento lógico si ayuda.
- Justificar con cuatro espacios.
- No utilizar *elseif*.
- Utilizar siempre llaves aunque el bloque sea solo de una línea y PHP no lo requiera.

```

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

```

Switch

- Usar un espacio antes y después de la expresión de control entre paréntesis, separando los operadores con espacios dentro de los paréntesis.
- Justificar con cuatro espacios.
- Utilizar siempre llaves aunque el bloque sea solo de una línea y PHP no lo requiera.
- El contenido bajo cada sentencia *case* debe estar justificado cuatro espacios adicionales.

```

• switch ($something) {
    case 1:
        break;

    case 2:
        break;

    default:
        break;
}

```

Foreach

Similar a los anteriores:

```

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

```

Operador ternario

- Solo se permite su uso en sentencias cortas de sencilla interpretación. Si no es así, utilizar una sentencia *if*.
- Utilizar espacios alrededor de los operadores para dejar claro cuándo tiene lugar cada operación.

```

$username = isset($user->username) ? $user->username : '';

```

Require/include

Cualquier fichero al que se pueda acceder desde el navegador debe incluir el fichero de configuración principal, **config.php**.

```
require(dirname(__FILE__) . '/../config.php');
```

Cualquier otro que se quiera incluir debe utilizar rutas absolutas a partir de *\$CFG->dirroot* o empezando con **__FILE__** o **__DIR__**, ya que las rutas relativas comenzando con *../* pueden provocar comportamientos extraños en PHP. Los scripts CLI (lanzados desde consola) no deben utilizar rutas relativas que comiencen con *../* a **config.php**.

La inclusión de archivos debería realizarse por lo general al comienzo del archivo o dentro de funciones o métodos que lo requieran.

El resto de scripts, con la excepción de librerías de terceros, para prevenir su ejecución directa que podría mostrar mensajes de error en servidores mal configurados, deberían incluir la línea:

```
defined('MOODLE_INTERNAL') || die();
```

DOCUMENTACIÓN Y COMENTARIOS

PHPDoc

Moodle permanece fiel tanto como es posible al formato estándar **PHPDoc** para documentar los archivos, clases y funciones. Facilita el uso de los IDEs por parte de los desarrolladores y permite generar documentación web automáticamente.

PHPDoc dispone de un catálogo de etiquetas que se pueden utilizar en diferentes ubicaciones. Hay definidas ciertas reglas a seguir en Moodle:

@copyright

Comprende el año y copyright del creador del fichero original. No cambiar en ficheros existentes.

```
@copyright 2008 Kim Bloggs
```


@license

Debe ser **GPL v3+** y usar este formato:

```
@license http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
```

@param

Usar nombre de tipo cortos (*bool* en lugar de boolean, *int* para integer) y no poner ningún guión o carácter similar a continuación del nombre de la variable, solo un espacio.

```
@param int $maximum The maximum allowed age for this nightclub
```

@return

- Esta etiqueta es obligatoria solo si la función tiene un sentencia **return**.
- Si la función no devuelve nada, “**return;**”, especificar *@return void*. Si devuelve **null**, “**return null**”, *@return null*.
- La descripción es opcional, pudiendo omitirse si la función es sencilla y ya se ha descrito que devuelve.
- Si puede devolver varios tipos de valores, enumerarlos separados por |, sin espacios.

```
@return string|bool If $output was true then return a string, else it will echo the string and return true.
```

@uses

Si una función utiliza las sentencias **die** o **exit**, añadir esta etiqueta al bloque de código para que los desarrolladores tengan presente que esta función puede finalizar la página.

```
@uses exit
```

@access

Especifica control de acceso a un elemento:

- Solo se debe usar cuando la definición del método no especifique el control de acceso.
- En el caso de funciones, evitar especificar acceso público ya que es redundante.

@package

Emplear siempre para etiquetar archivos PHP con el correcto nombre de componente *Frankenstyle*:

- Si el fichero es parte del componente de una extensión, utilizar el nombre del mismo (**mod_quiz** o **gradereport_xls**).
- Si el fichero es parte de un subsistema del núcleo, emplear *core_xxxx* where *xxxx* es el nombre definido en *get_core_subsystems()* (**core_enrol** o **core_group**).
- Si el fichero es uno de los pocos ficheros del núcleo que no pertenece a ningún subsistema, utilizar solo **core**.
- Cada fichero solo puede formar parte de un paquete.

No se define ninguna directriz para el uso de la etiqueta *@subpackage*. Utilizar libremente dentro de cada *@package*.

```
@package gradereport_xls
```

@category

Únicamente se utilizar para resaltar las clases públicas, funciones o ficheros que son parte de alguna de las APIs del núcleo, o que son un buen ejemplo de implementaciones de las mismas. El valor debe ser alguno de los especificados en la página de las APIs del núcleo.

```
@category preferences
```

@since

Al añadir una nueva clase o función a las librerías del núcleo de Moodle (o un nuevo método a una clase existente), utilizar esta etiqueta para documentar en qué versión de Moodle se añadió.

Por ejemplo:

```
@since Moodle 2.1
```

@see

Para dirigir al usuario a un elemento relacionado (página, clase, función, etc.).

```
@see some_other_function()
```

@link

Para dirigir al usuario a una URL externa:

```
@link http://docs.moodle.org/dev/Core\_APIs
```

También puede utilizarse para referir a algún apartado en la propia documentación, por ejemplo la descripción de una función. En estos casos se puede utilizar una etiqueta *@link* embebida (junto a una URL o el nombre de un elemento):

```
/**
 * This function uses {@link get_string()} to obtain the currency names...
 * .....
```

@deprecated

Cuando una API queda obsoleta, utilizar esta etiqueta para indicar a partir qué versión de Moodle lo es, y añadir las etiquetas *@todo* y *@see* si es posible. Asegurarse de especificar incidencias relacionadas (MDL-xxxx).

```

/**
 *
 * @deprecated since Moodle 2.0 MDL-3145 - not use this function more.
 * @todo MDL-22334 This will be deleted in Moodle 2.2.
 * @see class_name::new_function()
 */

```

Es importante que los desarrolladores actualicen su código. Valorar la posibilidad de añadir una llamada `debugging(..., DEBUG_DEVELOPER)`; parar repetir la advertencia de desfase. Si la función obsoleta no es ya soportada, lanzar una excepción `coding_exception`. Hay varios ejemplos en `lib/deprecatedlib.php`.

@throws

Indica que la función o método lanza una excepción. Alerta a los desarrolladores de que es posible que tengan que manejar excepciones.

Archivos

Todos los archivos con código PHP deben incluir, sin líneas en blanco después de la etiqueta de inicio de PHP, la licencia GPL completa al inicio, más un bloque de documentación separado conteniendo:

- Descripción corta en una línea del fichero
- Descripción más detallada
- Etiqueta `@package` (obligatoria)
- Etiqueta `@category` (solo si todo el contenido del fichero está relacionado con una de las APIs del núcleo)
- Etiqueta `@copyright` (obligatoria)
- Etiqueta `@license` (obligatoria)

```

<?php
// This file is part of Moodle - http://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify

```

```

// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle. If not, see <http://www.gnu.org/licenses/>.

/**
 * This is a one-line short description of the file.
 *
 * You can have a rather longer description of the file as well,
 * if you like, and it can span multiple lines.
 *
 * @package   mod_mymodule
 * @category  backup
 * @copyright 2008 Kim Bloggs
 * @license   http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
 */

```

Classes

Si la clase es el único contenido de un archivo se comenta con el bloque de documentación del archivo, siendo opcional incluir el bloque de documentación de la clase. Sin embargo, el resto de clases debe incluir un bloque de documentación como el siguiente:

```

/**
 * Short description for class.
 *
 * Long description for class (if any)...
 *
 * @package   mod_mymodule
 * @copyright 2008 Kim Bloggs
 */

```

```
* @license http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
*/
```

Añadir la etiqueta `@category` si se trata de un clase fundamental de las APIs del núcleo.

Propiedades

Todas las propiedades deberían ir precedidas de un bloque de documentación con información mínima:

```
class example {
    /** @var string This variable does something */
    protected $something;
}
```

o

```
class example {
    /**
     * This variable does something and has a very long description
     * which can wrap on multiple lines
     * @var string
     */
    protected $something;
}
```

Aunque haya varias propiedades con características comunes, no utilizar plantillas de bloques de documentación. En su lugar, documente cada propiedad explícitamente:

```
class zebra {
    /** @var int The number of white stripes */
    protected $whitestripes = 0;

    /** @var int The number of black stripes */
    protected $blackstripes = 0;
}
```

```
/** @var int The number of red stripes */
protected $redstripes = 0;
}
```

Constantes

Las constantes de una clase deberían documentarse como en el siguiente ejemplo:

```
class sam {
    /**
     * This is used when Sam is in a good mood.
     */
    const MOOD_GOOD = 0;
}
```

Funciones

Todas las funciones y métodos deberían tener un bloque de código como el siguiente:

```
/**
 * The description should be first, with asterisks laid out exactly
 * like this example. If you want to refer to a another function,
 * use @see as below. If it's useful to link to Moodle
 * documentation on the web, you can use a @link below or also
 * inline like this {@link http://docs.moodle.org/dev/something}
 * Then, add descriptions for each parameter and the return value as
 * follows.
 *
 * @see clean_param( )
 * @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 bool A status indicating success or failure
 */
```

Definiciones

Todas las definiciones debería documentarse tal como:

```

/**
 * PARAM_INT - integers only, use when expecting only numbers.
 */
define('PARAM_INT', 'int');

/**
 * PARAM_ALPHANUM - expected numbers and letters only.
 */
define('PARAM_ALPHANUM', 'alphanum');

```

Comentarios en línea

Los comentarios en línea debe utilizar el formato “// “(dos barras y un espacio), procurando ajustarlos o alinearlos con el código. Se recomienda que comiencen con letra mayúscula y uso correcto de símbolos de puntuación.

```

function forum_get_ratings_mean($postid, $scale, $ratings = null) {
    if (!$ratings) {

        $ratings = array(); // Initialize the empty array.

        $rates = $DB->get_records('forum_ratings',
                                array('post' => $postid));

        // Process each rating in turn.
        foreach ($rates as $rate) {
            ...
        }

        // Do something else here.
        something_else();
    }
}

```

Si el comentario está relacionado con alguna incidencia, añadir el correcto identificador, MDL-12345, en el comentario.

Usando TODO

Es especialmente importante si se tiene conocimiento de alguna incidencia que persiste en el código que debe ser tratada. Utilizar un TODO junto con un código MDL para marcar este punto.

```
// TODO MDL-12345 This works but is a bit of a hack and should be
//                               revised in future.
```

EXCEPCIONES

Las excepciones se utilizan para informar de errores, especialmente en el código de las librerías.

Lanzar una excepción tiene casi el mismo efecto que una llamada a *print_error*, pero es más flexible. Por ejemplo, se puede capturar y manejar de forma más controlada. Además facilita la escritura de pruebas de unidad.

Cualquier excepción no capturada lanzará la correspondiente llamada a *print_error*, informando al usuario del error.

No conviene abusar del uso de excepciones para manejar el flujo normal del código. Solo deberían utilizarse para gestionar situaciones erróneas.

Clases de excepciones

Hay definido un conjunto de excepciones personalizadas. La clase base es *moodle_exception*. Los argumentos pasados a *new moodle_exception(...)* son muy similares a los que se pasan a la llamada *print_error*.

Existen más subclases de las aquí listadas para tipos de error concretos. Para obtener la lista completa de tipos de excepciones, buscar por la expresión regular “*class +|w+_exception +extends*” o utilizar el IDE para listar todas las subclases de *moodle_exception*. Cuando sea apropiado, se deberían crear nuevas subclases para usar en el código.

Algunos tipos de excepciones destacados son:

- **moodle_exception**: clase base para las excepciones en Moodle. Utilizar cuando no hay un tipo más específico no es apropiado.
- **coding_exception**: lanzada cuando el error parece ser causado por un error del desarrollador. Normalmente lanzada por el código del núcleo que interactúa con

extensiones. Si se lanza una excepción de este tipo, conviene que el mensaje del error sea significativo para el autor de la extensión, de tal forma que pueda corregir el mismo.

- **dml_exception (y subclases)**: lanzada cuando falla una consulta a la base de datos.
- **file_exception**: lanzada por la API de ficheros.

FUNCIONES Y CONSTRUCTORES PELIGROSOS

PHP incluye varias características cuestionables que se desaconsejan enérgicamente, ya que frecuentemente son la causa de serios problemas de seguridad.

- no usar la función *eval()* – excepto en los paquetes de idiomas
- no utilizar *preg_replace()* con el modificador */e* – usar llamadas de retorno para prevenir ejecución de código PHP no intencionado
- no utilizar comillas inversas para la ejecución de comandos del shell

CÓDIGO SQL

En esta sección se dan recomendaciones sobre el estilo de código para consultas de base de datos complejas.

Las consultas completas se utilizan en los métodos *\$DB->get_records_sql()*, *\$DB->get_recordset_sql()* o *\$DB->execute()*. Los fragmentos de código SQL se pueden utilizar en métodos DML con el sufijo *_select()*.

Reglas generales

- Utilizar parámetros de sustitución.
- Todas las palabras clave de SQL en mayúsculas.
- Todas las consultas y fragmentos SQL encerrados en comillas dobles.
- Las consultas SQL complejas se escriben en varias líneas.
- Las consultas SQL en varias líneas se alinean a la derecha en base a los comandos *SELECT*, *FROM*, *JOIN*, *WHERE*, *GROUP BY* y *HAVING*.
- Utilizar *JOIN* en lugar de *INNER JOIN*.
- No utilizar combinaciones externas de tabla derecha (*right joins*).

- Emplear siempre *AS* para definir alias de columnas.

Comillas dobles

Todas las consultas y fragmentos SQL debe encerrarse en comillas dobles, no concatenando varios fragmentos SQL si es posible. Las comillas simples se utilizan en las cadenas sql, siendo útiles en el resaltado visual y en el autocompletado de código SQL en algunos IDEs.

```
$records = $DB->get_records_select('some_table', "id > ?",array(111));
```

Parámetros de sustitución

Todos los parámetros variables en una consulta debe especificarse utilizando parámetros de sustitución. Es posible utilizar tres tipos diferentes: *:named*, *?* y *\$!*. Se recomienda utilizar parámetros con nombre si hay más de un parámetro.

```
$sql = "SELECT *  
        FROM {some_table}  
        WHERE id > :above";  
$records = $DB->get_records_sql($sql, array('above'=>111));
```

Justificación

Respetar la justificación ilustrada en el siguiente ejemplo:

```
-----  
$sql = "SELECT a.id, a.value, a.another, b.id AS bid,  
             c.id AS cid, c.value AS yetanother  
        FROM {tablea} a  
        JOIN {tableb} b ON b.refa = a.id  
        LEFT JOIN {tablec} c ON c.refc = b.id  
        WHERE a.value = :avalue  
             AND b.value = :bvalue  
             AND c.value <> :cvalue";  
-----
```

Subconsultas

No hay reglas definidas para la justificación de las subconsultas; el factor a tener en cuenta es la facilidad para leer el código.

ANEXO V: GUÍA DE SEGURIDAD

Este anexo es un extracto de la guía (Moodle, 2014), donde se describe como escribir código seguro en Moodle. Referirse a la citada guía para obtener información más detallada.

Esta guía está organizada en base a los tipos de vulnerabilidades más comunes. Para cada uno se explica:

1. cuál es el peligro,
2. cómo está diseñado Moodle para evitarlo,
3. qué es necesario tener en cuenta como desarrollador para escribir código seguro y
4. qué hacer como administrador para que Moodle sea más seguro

SEGURIDAD EN LAS APLICACIONES WEB

Requisitos de aplicaciones web seguras

Algunas empresas requieren un nivel de seguridad máximo en las aplicaciones web. Las recomendaciones generales son:

- un entorno de administración separado
- la información delicada no se almacena en una aplicación web
- comunicación encriptada usando SSL
- registrar todas las acciones de los usuarios
- las aplicaciones en el servidor tienen que estar completamente separadas
- no permitir que los usuarios suban ficheros en el servidor
- no permitir que los usuarios puedan introducir texto enriquecido en el servidor
- validar la identidad del usuario y sus acciones en diferentes canales
- mantener el software actualizado
- no se recomiendan extensiones en los navegadores de terceros

- usa solo una página web, no abrir múltiples ventanas con diferentes páginas; abrir y cerrar el navegador antes y después de utilizar una aplicación segura

Equilibrio entre seguridad y funcionalidad

Muchas aplicaciones web actuales violan algunas de las reglas de diseño enumeradas: uso de texto enriquecido, ficheros adjuntos, información confidencial, etc. De hecho, el concepto de Web 2.0 es directamente contrario a estas directrices de seguridad.

Cuando se diseñan aplicaciones web, es necesario identificar qué necesidades tienen los usuarios y encontrar un equilibrio razonable entre funcionalidad y seguridad.

DISEÑO DE SEGURIDAD DE MOODLE

La seguridad de las aplicaciones web depende del uso que le vayan a dar los usuarios y las funcionalidades disponibles para cada tipo de usuario.

Tipos de usuarios

Administradores

Tienen los siguientes privilegios:

- cambiar la configuración
- crear cursos
- acceder a todos los cursos
- modificar los paquetes de idiomas
- modificar cualquier usuario

Indirectamente, los administradores pueden ejecutar código desde línea de comandos y código PHP. Se pueden limitar los permisos de los administradores de Moodle configurando *config.php* manualmente. Los administradores a bajo nivel del servidor no pueden ser controlados puesto que pueden acceder al contenido de los ficheros PHP y modificarlo.

Todos los administradores tienen que ser de confianza.

Profesores

Se dedican a la creación de contenidos, matriculación de estudiantes y a la enseñanza. Necesitan los siguientes privilegios:

- subir ficheros y enviar texto en formato html
- crear y gestionar actividades
- acceder a la puntuación e información personal de los alumnos

Subir ficheros utilizando Javascript o Flash se considera un riesgo de seguridad, pero no puede eliminarse esta funcionalidad.

Los profesores deben ser de confianza, no debe facilitarse acceso como profesores a los estudiantes. En teoría, los profesores pueden utilizar ataques XSS para obtener acceso como administradores.

Técnicamente se puede crear un sistema donde los profesores no puedan atacar a otros usuarios, pero sería necesario evitar el uso de Javascript, Flash, SCORM, formularios HTML, etc.

Estudiantes

Los estudiantes participan en los cursos y no son de confianza. Necesitan los siguientes privilegios:

- enviar texto formateado con imágenes y adjuntos
- subir documentos en formato binario

Los archivos subidos no deben abrirse directamente en el navegador desde el servidor. En su lugar, los archivos deben servirse desde otro dominio, o el servidor debe forzar la descarga de todos los archivos al disco duro local antes de abrirlos.

Todo el texto enviado por los alumnos debe ser depurado antes de mostrar el texto en cualquier página, para prevenir los ataques XSS a otros usuarios y el uso de scripts Flash, Javascript, SVG y HTML.

Invitados

Por razones de seguridad, los usuarios no registrados no pueden subir archivos ni texto a almacenar en la base de datos. Los invitados puede realizar spam sobre a otros usuarios o explotar errores en la limpieza de código, aprovechar otras vulnerabilidades o utilizar ataques de ingeniería social.

Riesgos de las capacidades

Moodle es un sistema muy flexible, donde los administradores pueden definir múltiples roles. Cada rol es un conjunto de capacidades definidas a nivel de sistema, aunque los roles se pueden modificar sobrescribiendo sus valores en contextos a más bajo nivel. Los riesgos forman parte de la descripción de cada capacidad, por lo que los administradores deben estar seguros de que solo usuarios de confianza obtienen capacidades potencialmente peligrosas.

RESUMEN DE LAS DIRECTRICES DE SEGURIDAD

Solo se tienen en cuenta aquellas que afectan al desarrollo de las extensiones de este proyecto. La ejecución de scripts desde línea de comandos no requiere ningún tipo de autenticación, dando por hecho que solo una persona con privilegios puede acceder a la consola. Por esta razón no se tienen en cuenta las recomendaciones en cuanto a autenticación de usuarios y control de privilegios.

Al no ejecutarse desde el navegador, también se obvian las directrices relacionadas con la entrada y salida de texto.

Para evitar problemas de control de acceso, deberían ejecutarse los scripts como el propietario del proceso del servidor web.

Escapar los datos antes de almacenar en la base de datos

Usar la librería XMLDB para evitar problemas de esta índole.

Cuando sea necesario utilizar código SQL personalizado, utilizar parámetros para insertar los valores en las sentencias.

Registrar cada acción

Los scripts deben utilizar la función **add_to_log** para registrar las acciones que se realicen durante su ejecución.

Otras recomendaciones

Estructurar adecuadamente el código, minimizando el uso de variables globales. Esto facilita verificar el flujo de los datos, y por lo tanto la seguridad.

Inicializar los objetos ($\$x = new stdClass$) y arrays ($\$x = array()$) antes de usarlos por primera vez.

ANEXO VI BASE DE DATOS: GUÍA DE ESTILO Y XMLDB.

Documentación extraída de (Moodle, s.f.).

GUÍA DE ESTILO

Para facilitar la creación de tablas que cumplan con las directrices de Moodle, se recomienda el uso del editor XMLDB integrado (*Administración del sitio – Desarrollo – Editor XMLDB*).

- Cada tabla debe tener un campo id (INT 10) autoincremental como clave primaria.
- La tabla principal que contiene las instancias de cada módulo (como ejemplo modulo) debe tener el mismo nombre del módulo y contener, como mínimo, los siguientes campos:
 - id – clave primaria descrita anteriormente
 - course – el id del curso al que pertenece la instancia
 - name – nombre completo de cada instancia del módulo
- Cada tabla asociada a un módulo que contiene información adicional, por ejemplo “cosas”, debe llamarse modulo_cosas (observar que se usa el plural).
- Las tablas principales en general tienen nombre simples en singular, y los nombres compuestos solo plural en la segunda palabra. Excepción a esta regla son las palabras reservadas, como “files”.
- Las tablas y columnas deben evitar utilizar palabras reservadas en cualquier base de datos. Los nombres de tabla pueden ser de hasta 28 caracteres y los de columna hasta 30.
- Los nombres de columna en minúscula, simples y cortos, siguiendo las mismas reglas que para los nombres de las variables.
- Cuando sea posible, las columnas que referencian el campo id de otra tabla, por ejemplo modulo, deben llamarse *moduloid*.

- Los campos booleanos deben implementarse como campos small integer (INT 4), conteniendo 0 ó 1.
- La mayoría de las tablas deberían tener un campo *timemodified* (INT 10) que se actualiza con la marca de tiempo obtenida con la función *time()* de PHP.
- Definir siempre un valor por defecto significativo para cada columna.
- Todos los nombres de las tablas tienen que empezar con el prefijo de la base de datos (*\$CFG->prefix*).
- Para garantizar la compatibilidad entre las bases de datos:
 - No usar **AS** para los alias de tabla
 - No usar nunca alias de tabla en las sentencias **DELETE**
 - Usar **AS** para los alias de columna
- No crear *UNIQUE KEYS* nunca. En su lugar utilizar *UNIQUE INDEXes*.
- Las tablas asociadas con un bloque debe seguir esta convención de nombre: *\$CFG->prefix + "block_" + nombre_del_bloque + cualquier_cosa*.
- Al referenciar variables enteras en consultas SQL, no encerrarlas entre comillas simples.
- Nunca usar comillas dobles para los valores de las variables en las consultas SQL.
- No utilizar Vistas en las bases de datos, Moodle no las soporta.

EDITOR XMLDB

XMLDB es la capa de abstracción de base de datos de Moodle. Es la librería de código que utiliza Moodle para interactuar y acceder a la base de datos. Para simplificar, en este apartado nos vamos a centrar en el uso del editor XMLDB. Más información sobre XMLDB en (Moodle).

El editor XMLDB es una herramienta para construir los ficheros .xml que especifican cómo Moodle debe crear y actualizar las tablas de la base de datos. Antes del uso de XMLDB, los desarrolladores tenían que crear ficheros independientes para trabajar con MySQL y Postgres.

Convierte la edición de tablas, campos, claves e índices en una tarea casi trivial. Todos los ficheros *install.xml* en los directorios *db* de Moodle pueden editarse con unos pocos clics. Estos ficheros contienen toda la información necesaria para crear los diferentes objetos en cada SGBD soportado.

Para poder manejar estos ficheros, el servidor web necesita permiso de escritura a todos los directorios *db* que contengan ficheros *install.xml*.

Creación/edición de tabla

1. En el directorio de la extensión, crear un directorio *db*, con permiso de escritura para el usuario del servidor web.
2. En Moodle, acceder a *Administración del sitio – Desarrollo – Editor XMLDB*.
3. En la lista de ubicaciones se muestra el nombre de la extensión, y el enlace *Crear* está habilitado.
4. Pulsar *Crear*.
5. Pulsar *Cargar* (carga el contenido del archivo *install.xml* en memoria).
6. Pulsar *Editar*.
7. Crear o editar las tablas de la extensión.
8. Una vez terminado, pulsar *Volver* y *Volver al principal*. En el listado de ubicaciones, pulsar *Guardar* en la fila de la extensión modificada.

Siguiendo estos pasos se habrá creado el fichero *install.xml*, el cual se utilizará para crear las tablas en la base de datos en conjunción con el fichero *version.php*.

Tras la instalación inicial de una extensión, para posteriores actualizaciones de la estructura de la tabla de la base de datos es necesario crear manualmente un fichero llamado *upgrade.php*, en la carpeta *db* del módulo. El esqueleto de este archivo será similar a este:

```
<?php
function xmldb_mymodule_upgrade($oldversion) {
    global $CFG;

    $result = TRUE;

    // Insert PHP code from XMLDB Editor here

    return $result;
}
?>
```

El código PHP se puede obtener accediendo al editor XMLDB, cargando el fichero *install.xml* correspondiente y pulsando el enlace *Ver código PHP*.

ANEXO VII: MANUALES DEL SISTEMA

MANUAL DE INSTALACIÓN

El código fuente está separado en tres carpetas:

- local/sinculpgc: extensión local;
- auth/casulpgc: extensión de identificación;
- enrol/sinculpgc: extensión de matriculación.

Cada una de las carpetas debe copiarse en la carpeta correspondiente del código fuente de Moodle, dependiendo del tipo de extensión que implementan. Es decir:

- local/sinculpgc (local) en la carpeta **local**;
- auth/casulpgc en la carpeta **auth**;
- enrol/sinculpgc (matriculación) en la carpeta **enrol**.

Seguidamente es necesario autenticarse como administrador en la plataforma y acceder a la página *Administración del sitio -> Notificaciones*, donde se iniciará el proceso de instalación de las nuevas extensiones. Basta con pulsar el botón *Actualizar base de datos* y la instalación finalizará en cuestión de segundos.

Para que la sincronización sea un proceso desatendido, se recomienda añadir una línea en el cron de un servidor donde está instalado el código fuente de Moodle, ejecutando el script de la extensión local **sincro.php**. La línea en el cron podría ser como la siguiente:

```
00 03 * * * /usr/bin/php /path/to/moodle/local/sinculpgc/cli/sincro.php  
>/dev/null
```

Para finalizar la instalación hay que añadir los valores de configuración de la base de datos externo a la variable global **\$CFG**, en el fichero *config.php*, antes de la llamada a *lib/setup.php*:

```
$CFG->plataforma = 'tp';  
$CFG->curso_academico = '201415';  
$CFG->ulpgcdbtype = 'oci8';  
$CFG->ulpgcdbhost = 'host.database.es:port';  
$CFG->ulpgcdbuser = 'USUARIO';  
$CFG->ulpgcdbpass = 'PASSWORD';  
$CFG->ulpgcdbname = 'NAMESERVICE';  
$CFG->ulpgccharset = 'we8iso8859p1';
```

MANUAL DE USUARIO

Extensión de identificación

Activación y configuración de la extensión

Autenticándonos en el sitio Moodle con rol de administrador, en el bloque Administración, seleccionar *Extensiones - Identificación - Gestionar identificación*. En esta pantalla se muestran las distintas extensiones de identificación disponibles, pudiéndose activar pulsando sobre el icono con forma de ojo bajo la columna *Habilitar*. A partir de este momento, la extensión estará activa y disponible para asignar a los usuarios.

moodlepfc Español - Internacional (es) Usted se ha identificado como Admin Usuario (Salir)

NAVEGACIÓN

- Página Principal
- Área personal
- Páginas del sitio
- Mi perfil
- Mis cursos

MARCAS DEL ADMINISTRADOR

Marcar esta página

ADMINISTRACIÓN

- Ajustes de mi perfil
- Administración del sitio
 - Notificaciones
 - Registro
 - Opciones avanzadas
 - Usuarios
 - Cursos
 - Calificaciones
 - Insignias
 - Ubicación
 - Idioma
- Extensiones
 - Instalar módulos externos
 - Vista general de extensiones
 - Autenticación**
 - Gestionar identificación**
 - CAS ULPGC (SSO)
 - Identificación basada en Email
 - Cuentas manuales
 - No hay sesión
 - Bloques

Gestionar identificación

Plugins de identificación disponibles

Nombre	Usuarios	Habilitar	Arriba/Abajo	Configuración	Configuración del test	Desinstalar
Cuentas manuales	2			Configuración		
No hay sesión	0			Configuración		
Identificación basada en Email	0	<input type="checkbox"/>	↓	Configuración		Desinstalar
CAS ULPGC (SSO)	0	<input type="checkbox"/>	↑	Configuración		Desinstalar
Usar un servidor CAS (SSO)	0	<input type="checkbox"/>		Configuración		Desinstalar
Usar una base de datos externa	0	<input type="checkbox"/>		Configuración	Configuración del test	Desinstalar
Usar servidor FirstClass	0	<input type="checkbox"/>		Configuración		Desinstalar
Usar un servidor IMAP	0	<input type="checkbox"/>		Configuración		Desinstalar
Usar un servidor LDAP	0	<input type="checkbox"/>		Configuración		
Identificación de la Red Moodle ("Moodle Network")	0	<input type="checkbox"/>		Configuración		
Usar un servidor NNTP	0	<input type="checkbox"/>		Configuración		Desinstalar
Sin identificación	0	<input type="checkbox"/>		Configuración		Desinstalar
PAM (Pluggable Authentication Modules)	0	<input type="checkbox"/>		Configuración		Desinstalar
Servidor POP3	0	<input type="checkbox"/>		Configuración		Desinstalar
Usar un servidor RADIUS	0	<input type="checkbox"/>		Configuración		Desinstalar
Shibboleth	0	<input type="checkbox"/>		Configuración		Desinstalar
Identificación de Servicios Web ("Web Services")	0	<input type="checkbox"/>		Configuración		

Figura VII-1 Activación de extensión de identificación

En la misma página, se puede acceder a la configuración de la extensión pulsando el enlace **Configuración**. La configuración comprende tres apartados:

- **Configuración del servidor CAS:** parámetros de configuración del servidor. En el caso de la ULPGC se solicita expresamente que al cerrar sesión en el Campus Virtual se cierre también en CAS, por lo que la opción *CAS logout option* se establece a *Sí* y se añade la URL de destino tras el cierre de sesión en el campo *Alternative logout return URL*.
- **Script de sincronización del Cron:** hay que especificar que sucede en el Campus Virtual con los usuarios que son eliminados de la base de datos externa. Es posible ignorar la eliminación externa (*Mantener interna*), suspender la cuenta de usuario (*Suspender interna*) o eliminar la cuenta también en el Campus Virtual (*Borrado completo*). En la ULPGC se opta por suspender la cuenta temporalmente porque en numerosas ocasiones vuelven a tener acceso.
- **Mapeado de datos:** son campos opcionales que indican que modificaciones se pueden realizar sobre la información personal de los usuarios tanto en la base de datos externa como en la del Campus Virtual, además de permitir bloquearlos. Los datos locales se

pueden actualizar solo cuando se crean o también en cada acceso y los datos externos se podrían actualizar tras cada modificación en el Campus Virtual. La política de la ULPGC actualmente restringe la modificación de la base de datos corporativa desde el Campus Virtual.

moodlepfc Español - Internacional (es) Usted se ha identificado como Admin Usuario (Salir)

NAVEGACIÓN

- Página Principal
- Área personal
- Páginas del sitio
- Mi perfil
- Mis cursos

MARCAS DEL ADMINISTRADOR

Marcar esta página

ADMINISTRACIÓN

- Ajustes de mi perfil
- Administración del sitio
 - Notificaciones
 - Registro
 - Opciones avanzadas
 - Usuarios
 - Cursos
 - Calificaciones
 - Insignias
 - Ubicación
 - Idioma
- Extensiones
 - Instalar módulos externos
 - Vista general de extensiones
 - Autenticación
 - Gestionar identificación
 - CAS ULPGC (SSO)**
 - Identificación basada en Email
 - Cuentas manuales
 - No hay sesión
 - Bloques

CAS ULPGC (SSO)

This method uses a CAS server (Central Authentication Service) to authenticate users in a Single Sign On environment (SSO). You can also use a simple LDAP authentication. If the given username and password are valid according to CAS, Moodle creates a new user entry in its database, taking user attributes from LDAP if required. On following logins only the username and password are checked.

CAS server configuration

Hostname: Hostname of the CAS server
eg: host.domain.fr

Base URI: URI of the server (nothing if no baseUri)
For example, if the CAS server responds to host.domain.fr/CAS/ then cas_baseuri = CAS/

Port: Port of the CAS server

CAS protocol version: CAS protocol version to use

Language: Select language for authentication pages

Proxy mode: Select 'yes' if you use CAS in proxy-mode

CAS logout option: Select 'yes' if you want to logout from CAS when you disconnect from Moodle

Server validation: Select 'yes' if you want to validate the server certificate

Certificate path: Path of the CA chain file (PEM Format) to validate the server certificate

Alternative logout return URL: Provide the URL that CAS users shall be redirected to after logging out.
If left empty, users will be redirected to the location that moodle will redirect users to

Script de sincronización del Cron

Usuario externo: Especifique qué hacer con una cuenta de usuario interna durante sincronización

Figura VII-2 Configuración de la extensión de identificación

Redirección a Servicio de identificación centralizada

Una directiva de la ULPGC establece que la autenticación debe realizarse utilizando única y exclusivamente el formulario del Servicio de identificación centralizada. Para que Moodle se adapte a esta restricción, en la página *Gestionar identificación*, donde activamos anteriormente la extensión, hay que asignar al parámetro *URL de acceso alternativo* la URL del servidor CAS. Así, cada vez que se intente acceder a una URL del Campus Virtual sin estar autenticado, se redirigirá al citado formulario.

Extensión de matriculación

Activación de la extensión

Autenticándonos en el sitio Moodle con rol de administrador, en el bloque Administración, seleccionar *Extensiones - Matriculaciones - Gestionar plugins de matriculación*. En esta pantalla

se muestran las distintas extensiones de matriculación disponibles, pudiéndose activar pulsando sobre el icono con forma de ojo bajo la columna *Habilitar*. A partir de este momento, la extensión estará activa y disponible para asignar a los usuarios.

Esta extensión no requiere configuración adicional.

The screenshot shows the Moodle administration interface for managing enrollment plugins. The main content area is titled "Gestionar plugins de matriculación" and contains a table of available plugins. The "Sincronización ULPGC" plugin is highlighted with a red box, indicating it is the focus of the activation process. The table columns are: Nombre, Ejemplos / matriculaciones, Versión, Habilitar, Arriba/Abajo, Configuración, Configuración del test, and Desinstalar.

Nombre	Ejemplos / matriculaciones	Versión	Habilitar	Arriba/Abajo	Configuración	Configuración del test	Desinstalar
Matriculación manual	1 / 1	2014051200	<input type="checkbox"/>	↓	Configuración		
Acceso de invitados	1 / 0	2014051200	<input type="checkbox"/>	↑ ↓	Configuración		Desinstalar
Auto-matriculación	1 / 0	2014051200	<input type="checkbox"/>	↑ ↓	Configuración		Desinstalar
Sincronizar cohorte	0 / 0	2014051200	<input type="checkbox"/>	↑ ↓	Configuración		Desinstalar
Sincronización ULPGC	0 / 0	2014061601	<input checked="" type="checkbox"/>	↑			Desinstalar
Categorías de matrícula	0 / 0	2014051200	<input type="checkbox"/>		Configuración		Desinstalar
Base de datos externa	0 / 0	2014051200	<input type="checkbox"/>		Configuración	Configuración del test	Desinstalar
Archivo plano (CSV)	0 / 0	2014051200	<input type="checkbox"/>		Configuración		Desinstalar
Archivo IMS Enterprise	0 / 0	2014051200	<input type="checkbox"/>		Configuración		Desinstalar
Inscripciones LDAP	0 / 0	2014051200	<input type="checkbox"/>		Configuración		Desinstalar
Meta-enlace de curso	0 / 0	2014051200	<input type="checkbox"/>		Configuración		Desinstalar
Matriculaciones remotas MNet	0 / 0	2014051200	<input type="checkbox"/>		Configuración		Desinstalar
PayPal	0 / 0	2014051200	<input type="checkbox"/>		Configuración		Desinstalar

Por favor, seleccione todos los conectores requeridos y colóquelos en orden adecuado. Los cambios de la tabla superior se guardan automáticamente.

Figura VII-3 Activación de la extensión de matriculación