



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

“ANÁLISIS DE LA PRODUCTIVIDAD PARA AGREGAR COMPONENTES EN LAS HERRAMIENTAS LMS APLICADO AL DESARROLLO DE UN MÓDULO ORIENTADO A LA NUEVA GESTIÓN DE CALIFICACIONES DE LA ESPOCH.”

**TESIS DE GRADO**

**Previa obtención del título de:**

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**Presentado por:**

**ERIKA JACQUELINE VILLACRÉS VALVERDE**

**GEOVANNY EUCLIDES SILVA PEÑAFIEL**

**RIOBAMBA – ECUADOR**

**2013**

## AGRADECIMIENTO

*Mi eterno agradecimiento a **Dios** por haberme brindado el don de la vida, y acompañarme en cada paso que he dado.*

*A mis **amig@s** que de alguna manera supieron brindarme su apoyo en esta etapa de mi vida.*

*Erika Jacqueline Villacrés Valverde.*

*Al creador de todas las cosas, el que me ha dado fortaleza para continuar cuando a punto de caer he estado; por ello, con toda la humildad que de mi corazón puede emanar, agradezco primeramente a **Dios**.*

*A **mi madre**, por ser el pilar más importante y pesar de nuestra distancia física, siento que estás conmigo siempre y aunque nos faltaron muchas cosas por vivir juntos, sé que este momento hubiera sido tan especial para ti como lo es para mí.*

*A **mi padre**, por su apoyo incondicional sin importar nuestras diferencias de opiniones.*

*Son muchas las personas que han formado parte de mi vida profesional a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en mi corazón, sin importar en donde estén quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones.*

*Para ellos: Muchas gracias y que Dios los bendiga.*

*Geovanny Euclides Silva Peñafiel*

## DEDICATORIA

*A la persona que es el centro y la razón de mi existencia **Mi Madre Zoila Valverde** por haberme brindado una vida llena de felicidad con su amor, consejos y apoyo en mis aciertos y fracasos, por enseñarme a luchar para conseguir mis metas y con su ejemplo hacer de mí una mujer de bien*

*A mis amadas hermanas **Silvia y Jimena** por acompañarme en los momentos más especiales y a pesar de las circunstancias permanecer juntas.*

*A mi **Sobrino Sebastián** que con su cariño e inocencia alegra mi vida a diario inspirándome a seguir adelante.*

*Erika Jacqueline Villacrés Valverde.*

*Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor. A Dios*

*A **mi madre Margoth**. Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.*

*A **mi padre Patricio** quien con sus consejos ha sabido guiarme para culminar mi carrera profesional.*

*Geovanny Euclides Silva Peñañiel.*

**FIRMAS RESPONSABILIDAD Y NOTA**

<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
Ing. Iván Menes Camejo <b>DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRONICA</b>	.....	.....
Ing. Raúl Rosero <b>DIRECTOR DE LA ESCUELA DE INGENIERIA EN SISTEMAS</b>	.....	.....
Ing. Gloria Arcos <b>DIRECTORA DE TESIS</b>	.....	.....
Ing. Julio Santillán <b>MIEMBRO DEL TRIBUNAL</b>	.....	.....
Tlgo. Carlos Rodríguez <b>DIRECTOR DEL CENTRO DE DOCUMENTACIÓN</b>	.....	.....

## **RESPONSABILIDAD DEL AUTOR**

“Nosotros, Erika Villacrés y Geovanny Silva, somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis, el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”.

### **FIRMAS:**

---

Erika Jacqueline Villacrés Valverde

---

Geovanny Euclides Silva Peñafiel

## **INDICE DE ABREVIATURAS**

### **A**

**AICC:** Aviation Industry CBT (computer-based training) Committee

**AGLP:** Affero General Public License / Licencia Pública General de Affero

**ASTD:** Sociedad Americana para Capacitación y Desarrollo

**ATRC:** Adaptive Technology Resource Centre

**AVA:** Ambiente Virtual de Aprendizaje

### **C**

**CD-ROM:** Compact Disc - Read Only Memory

**CERDECAM:** Centro de Investigación y Desarrollo

**CLE:**

**CMS:** Content Management System / Sistemas de Gestión de Contenidos

**CMC:** Computer Mediated Communication

### **E**

**ECAM:** Instituto Superior de Ingeniería Belga

### **F**

**FSF:** Free Software Foundation

### **G**

**GLP:** Licencia Pública General

### **H**

**HTTP:** Hypertext Transfer Protocol / Protocolo de Transferencia de Hipertexto

## **I**

**ICT:** Information and Communication Technology

**IMS:** Instructional Management System

**IIS:** Internet Information Services

**IEEE:** Institute of Electrical and Electronics Engineers / Instituto de Ingenieros Eléctricos y Electrónicos.

## **L**

**LMS:** Learning Management System

**LCMS:** Learning Content Management System

**LSS:** Learning Support System

**LP:** Learning Platform

**LAMP:** Linux, Apache, MySQL y PHP

**LDAP:** Lightweight Directory Access Protocol

## **M**

**MLE:** Managed Learning Environment

**MAMP:** Mac OS X, Apache, MySQL, PHP

## **O**

**OKI:** Iniciativa de Conocimiento Abierto

**OE:** Online Education

**OSI:** Open Source Initiative

## **P**

**PHP:** Personal Home Page

## **R**

**RLO:** Reusable Learning Objects

**RSS:** Really Simple Syndication

## **S**

**SCORM:** Sharable Content Object Reference Model

**SOAP:** Simple Object Access Protocol / Protocolo de Acceso a objetos simples

**SGBD:** Sistema de Gestión de Base de Datos

**SQL:** Structure Query Language / Lenguaje de Consulta Estructurado

## **T**

**TIC:** Tecnologías de la Información y Comunicación

## **V**

**VLE:** Virtual Learning Environment

## **W**

**WCM:** Web Content Management / Gestores de Contenido Web.

**WAMP:** Windows, Apache, MySQL, PHP.

**WCAG:** Web Content Accessibility Guidelines.

**WEC:** World Wide Web Consortium.

## **X**

**XHTML:** Extensible Hyper Text Markup Language

**XML:** Extended Markup Language / Lenguaje de marcado extendido

## ÍNDICE GENERAL

AGRADECIMIENTO.....	2
DEDICATORIA .....	3
FIRMAS RESPONSABILIDAD Y NOTA.....	4
RESPONSABILIDAD DEL AUTOR.....	5
INDICE DE ABREVIATURAS .....	6
ÍNDICE GENERAL .....	9
INDICE DE FIGURAS .....	17
INDICE DE TABLAS .....	19
INTRODUCCIÓN .....	21
CAPÍTULO I: .....	23
MARCO REFERENCIAL .....	23
1.1. Antecedentes .....	23
1.2. Justificación del Proyecto de Tesis .....	25
1.2.1. Justificación Teórica. ....	25
1.2.2. Justificación Práctica .....	26
1.3. Objetivos .....	27
1.3.1. Objetivo General: .....	27
1.3.2. Objetivos Específicos: .....	27
1.4. Hipótesis.....	28
1.5. Métodos y Técnicas.....	28
1.5.1. Métodos.....	28
1.5.2. Técnicas.....	28
CAPÍTULO II .....	29
MARCO TEÓRICO.....	29
2.1. Aprendizaje Electrónico / E-learning ( Electronic Learning ).....	30
2.2. Definición de E-Learning .....	30
2.2.1. Ventajas E-Learning .....	31
2.2.2. Desventajas E-Learning.....	32
2.2.3. B-learning (Blended Learning).....	33
2.2.4. Aula Virtual.....	34

2.2.4.1. Concepto de aula virtual .....	34
2.2.4.2. Objetivo Del Aula Virtual.....	35
2.2.4.3. Elementos esenciales Del Aula Virtual .....	35
2.2.4.4. Características del Aula Virtual .....	35
2.2.4.5. Formatos De Presentación .....	39
2.2.4.6. Ventajas Del Aula Virtual.....	39
2.2.4.7. Desventajas De Las Aulas Virtuales .....	40
2.3. SISTEMAS DE ADMINISTRACIÓN DE APRENDIZAJE (LMS) .....	41
2.3.1. Funciones y Características de un LMS.....	43
2.3.2. Beneficios de los LMS .....	44
2.3.3. Herramientas .....	45
2.3.4. Evolución de los Sistemas de Gestión de Aprendizaje .....	46
2.3.5. Plataformas de Enseñanza Virtual Actuales .....	48
2.3.6. Plataformas de enseñanza virtual libres .....	49
2.3.6.1. Software Libre .....	50
2.3.6.2. Licencias de Software Libre .....	52
2.4. ANÁLISIS DE LAS PLATAFORMAS PARA EDUCACIÓN VIRTUAL MÁS REPRESENTATIVAS.....	53
2.4.1. Descripción de los parámetros de comparación.....	54
2.4.1.1. Madurez del producto.....	54
2.4.1.2. Personalización.....	55
2.4.2. Descripción de las Plataformas de educación virtual.....	56
2.4.2.1. Descripción de la plataforma Moodle.....	56
2.4.2.2. Descripción de la plataforma ATutor .....	60
2.4.2.3. Descripción de la Plataforma Dokeos.....	62
2.4.2.4. Descripción de la Plataforma Claroline .....	63
2.4.2.5. Descripción de la Plataforma Sakai .....	64
2.4.3. Análisis Comparativo. ....	66
2.4.3.1. Establecimiento de Parámetros.....	68
2.4.3.1.1. Madurez del producto.....	68
Indicador 1.1: Idiomas .....	68
Indicador 1.2: Plataforma.....	68
Indicador 1.3: Base de datos .....	68

Indicador 1.4: Servidor Web .....	69
Indicador 1.5: Lenguaje de Programación .....	69
2.4.3.1.1.1. Valoraciones .....	70
2.4.3.1.1.2. Interpretación.....	70
2.4.3.1.1.3. Calificación .....	71
2.4.3.1.1.4. Representación de Resultados .....	73
2.4.3.1.2. Personalización.....	74
Indicador 2.1: Personalización de Apariencia .....	74
Indicador 2.2: Niveles de acceso .....	74
Indicador 2.3: Documentación .....	74
Indicador 2.4: Estándares .....	75
Indicador 2.5: Personalización y Creación de nuevos módulos.....	75
2.4.3.1.2.1. Valoración .....	76
2.4.3.1.2.2. Interpretación .....	76
2.4.3.1.2.3. Calificación.....	78
2.4.3.1.2.4. Representación de Resultados.....	79
2.4.3.2. Puntajes alcanzados .....	80
2.4.3.3. Interpretación:.....	81
2.4.3.4. Análisis de Resultados .....	82
2.5. EL MODELO SCRUM.....	83
2.5.1. Origen.....	83
2.5.2. Introducción al modelo.....	84
2.5.3. Control de la evolución del proyecto .....	85
2.5.4. Revisión de las Iteraciones .....	85
2.5.5. Desarrollo incremental .....	85
2.5.6. Desarrollo evolutivo .....	85
2.5.7. Auto-organización .....	86
2.5.8. Colaboración .....	86
2.5.9. Visión general del proceso .....	86
CAPITULO III.....	88
ANÁLISIS DE LA PRODUCTIVIDAD EN LA AGREGACIÓN DE MÓDULOS.....	88
3.1. MOODLE.....	89
3.1.1. Arquitectura de Moodle.....	91

3.1.1.1. Entorno de Moodle .....	92
3.1.1.2. Base de datos .....	92
3.1.1.3. Sistema operativo.....	93
3.1.1.4. Servidor Web .....	93
3.1.1.5. Estructura de directorios .....	94
3.1.1.6. Código de Moodle .....	95
3.1.1.7. Base de datos de Moodle .....	96
3.1.1.8. Datos de Moodle.....	96
3.1.1.9. Actividades .....	98
3.1.1.10. Recursos .....	98
3.1.1.11. Bloques .....	99
3.1.1.12. Control de acceso, inicio de sesión y roles de usuario.....	100
3.1.1.12.1. Permisos .....	100
3.1.1.12.2. Capacidades .....	101
3.1.1.12.3. Roles.....	102
3.1.1.12.4. Instalación de Moodle.....	102
3.1.1.12.5. Guías de estilo para desarrolladores.....	102
3.1.1.12.5.1. Estilo del código.....	102
3.1.1.12.5.2. Reglas generales.....	102
3.1.1.12.5.3. Estilo del código fuente.....	104
3.2. DOKEOS .....	106
3.2.1. Estructura del código de Dokeos .....	107
3.2.2. El API de Dokeos .....	109
3.2.3. Estructura de la base de datos de Dokeos .....	109
3.2.4. Convenciones de código propuestas .....	110
3.2.4.1. Lenguaje .....	110
3.2.4.2. Espacios en blanco.....	110
3.2.4.3. Indentación .....	110
3.2.4.4. Llaves .....	110
3.2.4.5. Variables y funciones.....	111
3.2.4.6. Espacios de nombres (Namespaces) .....	111
3.2.4.7. Constantes.....	112
3.2.4.8. Clases.....	112

3.2.4.9. Comentarios.....	112
3.3. AMBIENTE DE PRUEBA I.....	113
3.3.1. Descripción de los módulos de prueba.....	113
3.3.2. Desarrollo de los módulos de prueba en Moodle.....	114
3.3.2.1. Módulo 1.....	114
3.3.2.2. Módulo 2.....	117
3.3.2.3. Módulo 3.....	121
3.4. AMBIENTE DE PRUEBA II: DOKEOS.....	124
3.4.1. Desarrollo de los módulos de prueba en Dokeos.....	124
3.4.1.1. Módulo 1.....	124
3.4.1.2. Módulo 2.....	127
3.4.1.3. Módulo 3.....	129
3.5. Determinación de parámetros de productividad para la comparación.....	131
3.5.1. Acceso a Base de Datos.....	132
3.5.2. Facilidad de Mantenimiento.....	133
3.5.3. Codificación.....	133
3.5.4. Reutilización.....	134
3.5.5. Seguridad.....	135
3.5.6. Instalación.....	135
3.6. Análisis Comparativo.....	135
3.6.1. Acceso a Base de Datos.....	137
Indicador 1.1: Soporte para múltiples Bases de Datos.....	137
Indicador 1.2: Manipulación con la base de datos.....	137
Indicador 1.3: Desempeño con la base de datos.....	138
3.6.1.1. Valoración.....	138
3.6.1.2. Interpretación.....	139
3.6.1.3. Calificación.....	140
3.6.1.4. Representación de Resultados.....	141
3.6.2. Facilidad de Mantenimiento.....	141
Indicador 2.1: Facilidad de manejo de la estructura del LMS.....	141
Indicador 2.2: Escalabilidad.....	142
Indicador 2.3: Tiempo requerido en comprender el código.....	142
Indicador 2.4: Versiones y estándares.....	142

3.6.2.1. Valoraciones .....	143
3.6.2.2. Interpretación .....	144
3.6.2.3. Calificación.....	145
3.6.2.4. Representación de Resultados.....	146
3.6.3. Codificación .....	146
Indicador 3.1: Adaptación de hojas de estilo .....	146
Indicador 3.2: Estructura de directorios de los componentes del LMS.....	146
Indicador 3.3: Facilidad en la emisión de errores .....	147
Indicador 3.4: Funcionalidad.....	147
Indicador 3.5 Tiempo de Desarrollo.....	148
3.6.3.1. Valoraciones .....	148
3.6.3.2. Interpretación .....	149
3.6.3.3. Calificación.....	150
3.6.3.4. Representación de Resultados.....	151
3.6.4. Reutilización.....	151
Indicador 4.1: Herencia de Clase Base .....	151
Indicador 4.2: Reutilización de código.....	152
Indicador 4.3: Tamaño de la aplicación.....	152
3.6.4.1. Valoraciones .....	153
3.6.4.2. Interpretación .....	153
3.6.4.3. Calificación.....	154
3.6.4.4. Representación.....	154
3.6.5. Seguridad.....	155
Indicador 5.1: Utilización de roles y capacidades .....	155
Indicador 5.2: Integridad de datos .....	155
3.6.5.1. Valoraciones .....	156
3.6.5.2. Interpretación .....	156
3.6.5.3. Calificación.....	156
3.6.5.4. Representación.....	157
3.6.6. Instalación .....	157
Indicador 6.1: Tiempo de Instalación .....	157
3.6.6.1. Valoraciones .....	158
3.6.6.2. Interpretación.....	158

3.6.6.3. Calificación.....	158
3.6.6.4. Representación.....	159
3.6.7. Puntajes Alcanzados.....	159
3.6.8. Diagrama General de Resultados.....	161
3.6.9. Interpretación .....	161
3.7. Análisis de Resultados.....	162
3.8. Comprobación de Hipótesis.....	163
3.8.1. Resultados Finales .....	163
CAPÍTULO IV: .....	166
DESARROLLO DEL NUEVO MODULO PROTOTIPO ORIENTADO A LA NUEVA GESTIÓN DE CALIFICACIONES EN LA ESPOCH .....	166
4.1. Ingeniería de la Información.....	167
4.1.1. Definición del Ámbito .....	167
4.2. Análisis de Requerimientos .....	168
4.2.2. Definición del problema .....	168
4.2.3. Alternativas de solución .....	168
4.2.4. Descripción General .....	169
4.2.5. Funciones del Módulo .....	169
4.3. Estudio de Factibilidad .....	169
4.3.1. Factibilidad técnica.....	169
4.3.2. Factibilidad operativa .....	170
4.4. Desarrollo del Módulo.....	170
4.4.1. Visión y Planificación .....	170
4.4.2. Análisis y Diseño.....	171
4.4.2.1. Definición de Usuarios del Modulo .....	171
4.4.2.2. Análisis del sistema .....	172
4.4.3. Ciclos de Desarrollo – SCRUM .....	174
4.4.3.1. Iteración 0 – Investigación .....	174
4.4.3.2. Iteración 1 – Php en Moodle .....	174
4.5. Pruebas .....	174
4.5.1. Pruebas Unitarias.....	174
4.5.2. Pruebas del Módulo.....	175
CONCLUSIONES .....	176

RECOMENDACIONES .....	178
RESUMEN .....	180
ABSTRACT .....	181
BIBLIOGRAFÍA .....	182
ANEXOS .....	187

## INDICE DE FIGURAS

### CAPITULO II

Ilustración II. 1: Sistemas de Gestión de Aprendizaje (LMS).....	43
Ilustración II. 2: Logotipo Moodle .....	56
Ilustración II. 3: Interfaz Moodle .....	57
Ilustración II. 4: Logotipo ATutor .....	60
Ilustración II. 5: Logotipo Dokeos .....	62
Ilustración II. 6: Logotipo Claroline.....	63
Ilustración II. 7: Logotipo Sakai.....	64
Ilustración II. 8: Resultado final Parámetro 1- Madurez del Producto .....	73
Ilustración II. 9: Resultado final Parámetro 1- Personalización y Creación.....	79
Ilustración II. 10: Representación Gráfica de Resultados Generales .....	81
Ilustración II. 11: Resultado final del Análisis .....	83
Ilustración II. 12: Estructura del desarrollo ágil. ....	84
Ilustración II. 13: Estructura central de SCRUM .....	85
Ilustración II. 14: Sprint del SCRUM.....	87

### CAPITULO III

Ilustración III. 1: Entorno Moodle (WAMP) .....	92
Ilustración III. 2: Directorios de Moodle .....	94
Ilustración III. 3: Organización de ficheros de moodledata en Moodle 2.3 .....	97
Ilustración III. 4: Actividades y Recursos instaladas por defecto en Moodle 2.3 .....	98
Ilustración III. 5: Bloques Instalados por defecto en Moodle 2.3 .....	99
Ilustración III. 6: Comentarios en Dokeos.....	112
Ilustración III. 10: Archivos y carpetas creadas para el Módulo1 .....	114
Ilustración III. 11: Módulo1- código1 .....	115
Ilustración III. 12: Módulo1-código2 .....	116
Ilustración III. 13: Módulo1-código3 .....	116
Ilustración III. 14: Instalación Módulo 1 .....	117
Ilustración III. 15: Modulo1 instalado.....	117
Ilustración III. 16: Archivos creados para Módulo2 .....	118
Ilustración III. 17: Módulo 2- Código1 .....	119
Ilustración III. 18: Módulo2-código2 .....	119
Ilustración III. 19: Instalación Modulo2 .....	120
Ilustración III. 20: Módulo2 Instalado.....	121
Ilustración III. 21: Archivos y carpetas del modulo3.....	121
Ilustración III. 22: Instalación Módulo3 .....	123
Ilustración III. 23: Actualización en la Base de Datos del Módulo3 .....	123
Ilustración III. 24: Módulo 3 Instalado en los Módulos Actividad de Moodle .....	123

Ilustración III. 25: Agregación del Módulo 3 a un Curso .....	124
Ilustración III. 26: Módulo 1-Código 1 .....	124
Ilustración III. 27: Creación carpeta Modulo 1 .....	125
Ilustración III. 28: Modulo 1-Código 2 .....	125
Ilustración III. 29: Modulo 1-Código 3 .....	125
Ilustración III. 30: Modulo 1-Código 4 .....	126
Ilustración III. 31: Actualización nuevo módulo.....	126
Ilustración III. 32: Creación tabla para nuevo módulo.....	127
Ilustración III. 33: Modulo 2-Código 1 .....	127
Ilustración III. 34: Creación de carpeta para el nuevo módulo .....	128
Ilustración III. 35: Modulo 2-Código 2 .....	128
Ilustración III. 36: Modulo 2-Código 3 .....	129
Ilustración III. 37: Definición del componente 3.....	129
Ilustración III. 38: Creación de la carpeta del módulo 3 .....	129
Ilustración III. 39: Módulo 3- Código 1 .....	130
Ilustración III. 40: Módulo 3- Código 2 .....	130
Ilustración III. 41: Representación de Resultados - Parámetro 1: Acceso a Base de Datos .....	141
Ilustración III. 42: Representación de Resultados Parámetro2 - Facilidad de Mantenimiento .....	146
Ilustración III. 43: Representación de Resultados- Parámetro 3: Codificación .....	151
Ilustración III. 44: Representación de resultados - Parámetro 4: Reutilización .....	154
Ilustración III. 45: Representación de resultados - Parámetro 5: Seguridad.....	157
Ilustración III. 46: Representación de Resultados - Parámetro 6: Instalación .....	159
Ilustración III. 47: Gráfica General de Resultados.....	161
Ilustración III. 48: Resultado Final del Análisis.....	165

#### **CAPITULO IV**

Ilustración IV. 1: Diagrama de Caso de Uso Utilización del Módulo .....	173
Ilustración IV. 2: Diagrama Secuencia Utilización Módulo .....	173

## INDICE DE TABLAS

### CAPITULO II

Tabla II. I: Posibles Herramientas .....	39
Tabla II. II: Principales LMS .....	49
Tabla II. III: Descripción del Parámetro 1- Madurez del Producto .....	54
Tabla II. IV: Descripción del Parámetro 2- Personalización.....	55
Tabla II. V: Valoración cualitativa y cuantitativa.....	66
Tabla II. VI: Escala de valoración cualitativa y cuantitativa para los indicadores .....	66
Tabla II. VII: Valoración del Indicador 1.1: Idiomas .....	68
Tabla II. VIII: Valoración del Indicador 1.2: Plataformas .....	68
Tabla II. IX: Valoración del Indicador 1.3: Base de Datos .....	69
Tabla II. X: Valoración del Indicador 1.4: Servidor web .....	69
Tabla II. XI: Valoración del Indicador 1.5: Lenguaje de Programación .....	69
Tabla II. XII: Resultados del Parámetro 1: Madurez del Producto .....	70
Tabla II. XIII: Representación de resultados .....	73
Tabla II. XIV: Valoración del Parámetro 2 - Indicador 2.1 Apariencia .....	74
Tabla II. XV: Valoración del Parámetro 2 - Indicador 2.2 Niveles de Acceso.....	74
Tabla II. XVI: Valoración del Parámetro 2 - Indicador 2.3 Documentación .....	75
Tabla II. XVII: Valoración del Parámetro 2 - Indicador 2.4 Estándares.....	75
Tabla II. XVIII: Valoración del Parámetro 2 - Indicador 2.5 Personalización y Creación .....	75
Tabla II. XIX: Valoración de los Indicadores del Parámetro 2– Personalización .....	76
Tabla II. XX: Representación de Resultados Parámetro 2 .....	79
Tabla II. XXI: Cuadro de Resultados Parámetro - Indicador .....	80
Tabla II. XXII: Resultados Generales .....	81
Tabla II. XXIII: Valores y Porcentajes Finales.....	82

### CAPITULO III

Tabla III. I: Parámetros e Indicadores a Valorar .....	131
Tabla III. II: Descripción Parámetro 1: Acceso a Base de Datos .....	132
Tabla III. III: Descripción Parámetro 2: Facilidad de Mantenimiento.....	133
Tabla III. IV: Descripción Parámetro 3: Codificación .....	133
Tabla III. V: Descripción Parámetro 4: Reutilización .....	134
Tabla III. VI: Descripción Parámetro 5: Seguridad .....	135
Tabla III. VII: Descripción Parámetro 6: Instalación .....	135
Tabla III. VIII: Valoración cualitativa y cuantitativa.....	136
Tabla III. IX: Escala de valoración cualitativa y cuantitativa para los indicadores .....	136
Tabla III. X: Valoración del Indicador 1.1 - Soporte para múltiples Bases de Datos .....	137
Tabla III. XI: Valoración del Indicador 1.2: Manipulación con la Base de Datos.....	138
Tabla III. XII: Valoración del Indicador 1.3: Desempeño con la Base de Datos .....	138

Tabla III. XIII: Resultados Parámetro 1: Acceso a Base de Datos .....	138
Tabla III. XIV: Valoración del Indicador 2.1.: Facilidad de manejo de la estructura del LMS.....	141
Tabla III. XV: Valoración del Indicador 2.2: Escalabilidad .....	142
Tabla III. XVI: Valoración del Indicador 2.3: Tiempo requerido en comprender el código .....	142
Tabla III. XVII: Valoración del Indicador 2.4: Versiones y Estándares .....	143
Tabla III. XVIII: Resultados Parámetro 2: Facilidad de Actualizaciones .....	143
Tabla III. XIX: Valoración del Indicador 3.1: Adaptación de hojas de estilo .....	146
Tabla III. XX: Valoración del Indicador 3.2: Estructura de los directorios de los componentes del LMS .....	147
Tabla III. XXI: Valoración del Indicador 3.3: Facilidad den la emisión de errores.....	147
Tabla III. XXII: Valoración del Indicador 3.4: Funcionalidad .....	147
Tabla III. XXIII: Valoración del Indicador 3.5: Tiempo de Desarrollo .....	148
Tabla III. XXIV: Resultados Parámetro 3: Codificación .....	148
Tabla III. XXV: Valoración del Indicador 4.1: Herencia de Clase Base .....	151
Tabla III. XXVI: Valoración del Indicador 4.2: Reutilización de Código .....	152
Tabla III. XXVII. Valoración del Indicador 4.3: Tamaño de la Aplicación .....	152
Tabla III. XXVIII. Resultados del Parámetro 4: Reutilización .....	153
Tabla III. XXIX: Valoración del Indicador 5.1: Utilización de roles y Capacidades .....	155
Tabla III. XXX: Valoración del Indicador 5.2: Integridad de Datos.....	155
Tabla III. XXXI: Resultados del Parámetro 5: Seguridad .....	156
Tabla III. XXXII: Valoración del Indicador 6.1: Tiempo de Instalación .....	157
Tabla III. XXXIII: Resultados del Parámetro 6: Instalación .....	158
Tabla III. XXXIV: Cuadro Resultados de Parámetro e Indicador .....	159
Tabla III. XXXV: Resultados Generales .....	161
Tabla III. XXXVI: Resultados finales.....	163
Tabla III. XXXVII: Valores y Porcentajes Finales .....	164

## **CAPITULO IV**

Tabla IV. I: iteraciones .....	171
Tabla IV. II: Caso de Uso Cálculo Parciales.....	172

## INTRODUCCIÓN

El presente trabajo de investigación de tesis previo a la obtención del título de Ingeniería en Sistemas Informáticos, trata sobre el “ANÁLISIS DE LA PRODUCTIVIDAD PARA AGREGAR COMPONENTES EN LAS HERRAMIENTAS LMS APLICADO AL DESARROLLO DE UN MÓDULO ORIENTADO A LA NUEVA GESTIÓN DE CALIFICACIONES DE LA ESPOCH”.

Es un estudio y análisis investigativo de los LMS, Moodle, Dokeos, ATutor, Claroline y Sakai que se ubican como los más populares en la actualidad a fin de obtener como resultado las 2 herramientas LMS y de las dos herramientas obtener la más productiva al momento de la agregación de componentes.

Como precedente al momento de agregar un nuevo componente personalizado resulta factible la utilización de un LMS que facilite la agregación pero esto puede resultar un poco confuso a la hora de decidirse por una determinada herramienta antes de empezar a desarrollar el componente, ya que nos enfrentamos a dos tipos de herramientas de muy similares características, basados en los mismos conceptos, y lo más importante garantizando, tanto Moodle, así como también Dokeos, la agregación de nuevos componentes personalizados.

El presente trabajo de investigación contiene los siguientes capítulos:

El en CAPÍTULO I se presenta el planteamiento de la investigación, antecedentes, hipótesis, métodos y técnicas, es todo el marco referencial para el desarrollo de la tesis.

En el CAPÍTULO II se detalla los aspectos teóricos motivo de la investigación, conceptos, terminologías, referenciadas al objeto de estudio. Y el análisis comparativo entre las 5 herramientas LMS más populares, Moodle, Dokeos, ATutor, Claroline y Sakai evaluándolas con los parámetros Madurez del producto y Personalización seleccionando a Moodle con el 90% y Dokeos con el 60% como las dos herramientas más aptas para la agregación de nuevos componentes con .

Continuando con el CAPÍTULO III se desarrolla el análisis de los LMS Moodle y Dokeos, agregando módulos prototipos de similares características en ambas herramientas, estableciendo parámetros o métricas de productividad en la agregación de nuevos componentes, tales como Acceso a la Base de Datos, Facilidad de Mantenimiento, Codificación , Reutilización, Seguridad e Instalación y así determinar cuál LMS brinda una mayor productividad al momento de agregar un componente, sometiendo a cada prototipo a diferentes escenarios de pruebas, finalizando con la demostración de la hipótesis.

Mediante el análisis realizado se obtuvo a Moodle como la herramienta más productiva en la agregación de componentes a Moodle con un 93.57% superando con un 13.93% a Dokeos que obtuvo un 79.64%.

En el CAPÍTULO IV se detalla la parte aplicativa de la tesis, contiene los requisitos de ingeniería de software, desarrollo del nuevo módulo o componente, estándares de desarrollo, todo referente al nuevo módulo prototipo orientado a la nueva gestión de calificaciones por parte de los docentes en la Escuela Superior Politécnica de Chimborazo.

## **CAPÍTULO I: MARCO REFERENCIAL**

### **1.1. Antecedentes**

Con aplicaciones basadas en las TIC, aparecen nuevos entornos de aprendizaje en los que es posible la comunicación, acción e interacción social de los agentes educativos, dando origen a una de las modalidades de educación a distancia conocida como E-Learning, medio de aprendizaje no presencial digital, en el cual el alumno accede al contenido a través de un computador o dispositivo móvil.

Un Ambiente Virtual de Aprendizaje (AVA) ó Virtual Learning Environment (VLE) es un sistema de software diseñado para facilitar a profesores la gestión de cursos virtuales para sus estudiantes, especialmente ayudándolos en la administración y desarrollo del curso. Se basan en el principio de aprendizaje colaborativo donde se permite a los estudiantes realizar sus aportes y expresar sus inquietudes en los foros, además van apoyados de herramientas multimediales que hagan más agradable el aprendizaje pasando de ser simplemente un texto en línea, a un entorno interactivo de construcción de conocimiento.

Tales sistemas e-learning a veces también son llamados:

- Sistemas de Gestión del Aprendizaje, del inglés Learning Management System (LMS)
- Sistema de Gestión de Curso (CMS, del inglés Course Management System)
- Sistema de Gestión de Contenido para el Aprendizaje (LCMS, del inglés Learning Content Management System)
- Ambientes de Aprendizaje Gestionado (MLE, del inglés Managed Learning Environment)
- Sistema de Apoyo al Aprendizaje (LSS, del inglés Learning Support System)
- Plataforma de Aprendizaje (LP, del inglés Learning Platform);
- Es un medio de educación llamado "comunicación mediante el computador" (CMC, del inglés Computer Mediated Communication)
- "educación en línea" (OE, del inglés online education)

Existe una amplia oferta de Ambientes Virtuales de Aprendizaje, entre la que cabe destacar las plataformas de uso libre o código abierto. Como por ejemplo Moodle, Dokeos, Sakai, Claroline, ATutor, etc., todas estas herramientas presentan similares características y conceptos, lo cual hace necesario un análisis de cada una de ellas para poder elegir las herramientas más adecuadas a nuestra necesidad.

Al resultar semejantes en características puede resultar un poco confuso a la hora de decidirse por una determinada herramienta antes del desarrollo y agregación de un componente, ya que las herramientas poseen módulos predeterminados para su agregación pero al momento de crear o desarrollar un nuevo componente personalizado por los desarrolladores existe desconocimiento en las características, funcionalidades, estructura, etc... nos enfrentamos a dos tipos de LMS muy similares y basados en los mismos conceptos.

Al personalizar nuevos componentes se puede establecer condiciones, restricciones de cierta manera el acceso a cierta información, configurar distintos tipos de vistas para los diferentes roles, establecer capacidades para los administradores, cambiar funcionalidades y aspectos generales añadiendo funcionalidad a la herramienta. Para

ello se requiere conocer un poco más a fondo la arquitectura y el funcionamiento interno de las mencionadas herramientas.

A la hora de desarrollar nuestro propio componente, también hay que tener en cuenta la estructura y una serie de normas de seguridad, es muy importante mantener la seguridad en estos procedimientos para poder garantizar la integridad de los datos manejados por el nuevo módulo.

La Escuela Superior Politécnica de Chimborazo, se encuentra en proceso de Acreditación y Calidad Universitaria, por lo cual ha implementado un nuevo sistema de evaluación estudiantil, dividiendo la nota acumulada 28 puntos en tres aportes, primer aporte sobre 8, segundo aporte sobre 10 puntos y tercer aporte sobre 10 puntos; lo cual cambia la temática de evaluación mantenida por algún tiempo de algunos docentes, lo cual les lleva a requerir cambios al entorno virtual.

## **1.2. Justificación del Proyecto de Tesis.**

### **1.2.1. Justificación Teórica.**

Debido a la integración de herramientas educativas bien estructuradas se hace necesaria la agregación de nuevos componentes los cuales permitan satisfacer necesidades educativas (docente - estudiante) para enriquecer, fortalecer y facilitar el aprendizaje.

La agregación de nuevos componentes debe contribuir al desarrollo y mejora de la herramienta LMS. Cada componente tiene una función específica y personalizada que dependerá de requerimientos propios de la persona que utilice la herramienta.

Hoy en día, existe una amplia oferta de Ambientes Virtuales de Aprendizaje, entre la que cabe destacar las plataformas de uso libre o código abierto. Los más reconocidos son Moodle, Dokeos, Eminus, Sakai, Claroline y ATutor cada uno con sus características particulares pero con un buen desarrollo de sus aplicaciones.

La presente investigación pretende realizar 2 análisis:

- De las herramientas LMS más populares del mundo educativo, en este análisis se seleccionaran en base a parámetros las herramientas más adecuadas a nuestra necesidad.
- Al ya contar con las herramientas más aptas para nuestro requerimiento nos encontraremos con el desconocimiento, con la igualdad de conceptos, con la semejanza de características, por lo cual se considera importante el poder determinar por medio de este análisis el comportamiento de estas herramientas para poder elegir con claridad la mejor opción al momento de agregar un componente.

### **1.2.2. Justificación Práctica**

Con el propósito de determinar las herramientas más adecuadas se realizara el análisis en base a las características que debe cumplir un Ambiente Virtual con el fin de poder seleccionar las más adecuadas.

En las herramientas seleccionadas agregaremos prototipos de componentes de iguales características, los cuales permitan evaluar mediante parámetros la agilidad en la agregación de nuevos componentes.

Con la situación actual en la que se encuentra la Escuela Superior Politécnica de Chimborazo ESPOCH a través de sus Docentes es necesario desarrollar un módulo prototipo adicional orientado a la gestión de calificaciones en el entorno virtual, el mismo que permitirá automatizar el proceso de evaluación virtual y se regirá al nuevo sistema de evaluación estudiantil, el cual contará con una interfaz amigable con el usuario. El módulo facilitará la administración de porcentajes equivalentes en cada evaluación, trabajo, tarea, etc.

El nuevo módulo prototipo a desarrollar permitirá obtener las notas resultantes de los diferentes aportes (foros, tareas, pruebas on-line, etc.) de los alumnos, realizar el cálculo correspondiente y dar la nota de los diferentes aportes divididos de la siguiente manera: primer aporte sobre 8, segundo aporte sobre 10 puntos y tercer aporte sobre 10 puntos dando un acumulado de 28 puntos.

### **1.3. Objetivos**

#### **1.3.1. Objetivo General:**

Realizar un análisis de la productividad para agregar componentes en las herramientas LMS aplicado al desarrollo de un módulo orientado a la nueva gestión de calificaciones de la ESPOCH.

#### **1.3.2. Objetivos Específicos:**

- Estudiar las características y beneficios que nos proporcionan las herramientas LMS.
- Seleccionar las principales herramientas LMS que permitan realizar la agregación de componentes para su análisis.
- Establecer parámetros de comparación los cuales nos permitan determinar con claridad que herramienta de las seleccionadas se adapta mejor a la agregación de un módulo prototipo orientado a la gestión de calificaciones.
- Seleccionar el LMS más adecuado para la agregación de componentes en base a los parámetros establecidos.
- Construir un módulo prototipo adaptado al nuevo sistema de evaluación (calificación) estudiantil por parte de los Docentes de la Escuela Superior Politécnica de Chimborazo.

#### **1.4. Hipótesis**

La agregación de módulos en Moodle es más productiva que los demás LMS analizados.

#### **1.5. Métodos y Técnicas**

##### **1.5.1. Métodos**

Para la comprobación de nuestra hipótesis será necesaria la aplicación de un método científico, que permitirá establecer una secuencia ordenada de acciones que nos llevarán a establecer las conclusiones sobre la productividad en la agregación de componentes en las herramientas LMS se empleará la metodología SCRUM.

##### **1.5.2. Técnicas**

Para lo que tiene que ver en cuanto a fuentes de información se utilizarán principalmente lo que se refieran al tema de investigación como la observación, libros, revistas, páginas web, etc. además, se contactará a especialistas en los temas mediante video-llamadas, foros en línea, etc. También se empleará la observación y análisis por parte de los investigadores para hacer deducciones e inducciones sobre el tema de tesis.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

En este capítulo se muestran y clarifican los aspectos teóricos en los que se fundamenta este proyecto. Primeramente se explicará en qué consiste el Aprendizaje Electrónico (E-Learning, B-Learning) características y ventajas. Las Herramientas LMS su historia, características y funcionalidades. Muchas instituciones educativas utilizan plataformas de enseñanza virtual, que integran las tecnologías que soportan la enseñanza virtual. Analizamos las cinco plataformas de software libre más populares y sus características indispensables de LMS y las elementales en la agregación de componentes. Se realizará un análisis comparativo determinando las dos herramientas LMS más populares.

## **2.1. Aprendizaje Electrónico / E-learning ( Electronic Learning )**

En la Actualidad está en auge el término de E- Learning, el cual es uno de los conceptos más utilizados y al mismo tiempo uno de los menos entendidos.

En esta sección se plasmarán los términos de e-learning y todos los aspectos relacionados a él, tales como características, ventajas, desventajas. B-Learning con sus características y una introducción y lo referente a Aulas Virtuales.

## **2.2. Definición de E-Learning**

Los sistemas o entornos e-learning son la modalidad más innovadora en la educación a distancia basada en Web, que hacen uso de los servicios y facilidades de Internet para hacer posible el proceso de enseñanza-aprendizaje.

Diferentes autores definen el concepto de e-learning de muy diversas formas, según la idiosincrasia y el ámbito de aplicación en el que se está utilizando. Si se toma como referencia la raíz de la palabra, e-learning se traduce como “aprendizaje electrónico”, y como tal, en su concepto más amplio puede comprender cualquier actividad educativa que utilice medios electrónicos para realizar todo o parte del proceso formativo.

Marcus S. Bowles define el Aprendizaje electrónico como un Aprendizaje que envuelve la adquisición, generación y transferencia de conocimientos usando tecnología de información y comunicación (ICT- Information and Communication Technology o TIC). [7]

En el glosario de términos de la American Society of Training and Development Kaplan-Leiserson, s/f se encuentra la siguiente definición:

**E-learning:** (Electronic Learning) Término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en Web, aprendizaje basado en computadora, aulas virtuales y colaboración digital. Incluye entrega de contenidos vía Internet,

intranet/extranet, audio y videgrabaciones, transmisiones vía satélite, TV interactiva, CD-ROM y más. [8]

Ésta es una definición que abre el rango del e-learning prácticamente a cualquier proceso relacionado con educación y tecnologías. Sin embargo, otros autores acotan más el alcance del e-learning reduciéndolo exclusivamente al ámbito de Internet, como Rosenberg, 2001 que lo define como: el uso de tecnologías de Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento. [20] Está basado en tres criterios fundamentales:

- El e-learning trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.
- Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.
- Se enfoca en la visión más amplia del aprendizaje que va más allá de los paradigmas tradicionales de capacitación.

### **2.2.1. Ventajas E-Learning [18]**

- La reducción de costos para dar cursos a más número de participantes que los de un aula de clases tradicional.
- Flexibilidad de horario esto es de gran importancia ya que permite a los estudiantes formar su calendarización de la mejor forma posible.
- Permite al estudiante despertar el interés de generar una interacción más óptima con los cursos ayudándolos a ser más activos por medio de los foros de discusión.
- Facilidad de acceso, el hecho de que el estudiante pueda seguir cualquier curso de e-learning, necesitando únicamente una conexión a Internet y una computadora personal o de escritorio con su respectivo navegador de Internet.
- Reducción de los tiempos de aprendizaje, porque se enseña lo esencial con ejemplos claros y precisos.

- Aumento de la retención, gracias a los medios multimedia y audiovisuales e interactivos con los cuales podría contar el curso.
- La personalización de la formación, los cursos e-learning ofrecen la ventaja de ser personalizados, el estudiante se identifica por medio de su usuario. Le aparece toda la información que el profesor haya dispuesto.
- Accesibilidad a los estudiantes discapacitados que no pueden transportarse continuamente a la universidad.
- Elimina las barreras de tiempo, distancia, ya que los estudiantes pueden tomar a su propio ritmo y decidir sobre su aprendizaje en su vida educativa.

### **2.2.2. Desventajas E-Learning**

- Miedo al cambio: porque se tiene la mentalidad de que para poder aprender hay que ir a sentarse a un salón de clases tradicional.
- Pobreza: muchos estudiantes universitarios cuentan con un nivel económico muy bajo, lo que les impide tener una computadora y acceso a internet en sus hogares.
- Motivación: si el curso no cuenta con un diseño agradable, la flexibilidad adecuada, el material que posee no está bien definido y elaborado el alumno podría optar por renunciar al curso.
- Retroalimentación: si el curso no cuenta con una debida retroalimentación por no hacer uso de los foros de discusión y chat para solventar las dudas caerá en la monotonía rutinaria de los cursos tradicionales.
- Alto costo de la elaboración del material didáctico: Requiere mucho tiempo elaborar el material didáctico para una clase virtual más que para una presencial.
- Se requiere capacitación continua de los profesores o tutores.
- El estudiante debe de ser más organizado y metódico para completar satisfactoriamente el curso en línea.

La impartición de un curso también puede llevarse a cabo de forma combinada, con una parte basada en e-learning y otra en los métodos tradicionales, dándose lo que se conoce como blended learning. Thorne, 2003. [29]

### 2.2.3. B-learning (Blended Learning)

La incorporación de las TICs en el aula ha supuesto un cambio en la enseñanza tradicional a nivel metodológico y actitudinal tanto para los profesionales de la enseñanza, como para los propios alumnos.

Cuando la instrucción tradicional en el aula es combinada con tecnología Web, se le conoce como “inserción de tecnología” o Mejoramiento en Web. Actualmente, es uno de los segmentos más populares y de más extenso crecimiento en la educación superior.

Recientemente esta innovación ha dado lugar a un nuevo modelo denominado b-learning (Blended Learning), donde no se trata solo de agregar tecnología a la clase, sino de reemplazar algunas actividades de aprendizaje con otra apoyadas con tecnología. Es decir, se trata de un modelo compuesto por instrucción presencial y características del aprendizaje electrónico o e-learning, con la finalidad de potenciar las fortalezas y disminuir las limitaciones de ambas modalidades.

El término “blended learning”, se puede traducir al castellano como Aprendizaje Mixto, sigue una tendencia con una marcada raíz procedente del campo de la psicología escolar en la que destaca el término “aprendizaje” como contrapuesto al de “enseñanza” .

Tanto el aprendizaje electrónico (e-learning) como el aprendizaje combinado (b-learning) tienen sus fundamentos en las teorías del aprendizaje y su aplicación al uso de medios tecnológicos. En este sentido, podríamos identificar las siguientes teorías:

- **Conductismo:** atención a ejercicios de tipo mecánico con retroalimentación inmediata (por ejemplo, tutoriales).
- **Constructivismo:** atención a la construcción de los conocimientos basado en el esfuerzo individual (por ejemplo, exploración en bibliotecas virtuales, estudio de casos).
- **Cognitivismo:** atención a las estrategias de aprender a aprender y capacidad indagativa de los estudiantes (por ejemplo, exploración).

- **Humanismo:** atención a diferencias individuales y al trabajo colaborativo (por ejemplo, estilos y ritmos de aprendizaje).

## **2.2.4. Aula Virtual**

### **2.2.4.1. Concepto de aula virtual**

Concepto que se ha venido desarrollando a partir de la década de los ochenta, éste término se le adjudica a ROXANNE HILTZ quien la define como “el empleo de comunicaciones mediadas por computadores para crear un ambiente electrónico semejante a las formas de comunicación que normalmente se producen en el aula convencional”. [28]

A través de éste entorno el alumno puede acceder y desarrollar una serie de acciones que son propias de un proceso de enseñanza presencial como conversar, leer documentos, realizar ejercicios, formular preguntas al docente, trabajar en equipo, etc. Todo ello de forma simulada sin que medie una interacción física entre docentes y alumnos.

Una “clase virtual es un entorno de enseñanza y aprendizaje inserto en un sistema de comunicación mediado por ordenador”. A través de este entorno el alumno puede acceder y desarrollar una serie de acciones que son las propias de un proceso de enseñanza presencial como conversar, leer documentos, realizar ejercicios, formular preguntas al docente, trabajar en equipo, etc. Todo ello de forma simulada sin que medie una interacción física entre docente y alumnos. [10]

El aula virtual es un espacio de acceso permanente a la información y material educativo necesarios para fortalecer el proceso de formación de los alumnos, ya que en ella encontrarán información detallada, específica y vasta acerca de un determinado curso a tiempo real, utilizando para ello la Red de Redes: Internet. La forma en la que las instituciones académicas pueden materializar la gestión del conocimiento es a través

de la enseñanza virtual porque permite colaborar, innovar y tener capacidad de respuesta ante los cambios que se están produciendo en la sociedad.

#### **2.2.4.2. Objetivo Del Aula Virtual**

- Aula Virtual tiene por objetivo el acompañar a las diferentes instituciones educativas y personas naturales que deseen incursionar en las nuevas tecnologías.
- Apoyar y orientar sus procesos de mejora a través de la asesoría e implementación de las soluciones más convenientes según su visión y misión institucional, como definiendo las estrategias de implementación eficaz.

#### **2.2.4.3. Elementos esenciales Del Aula Virtual**

Los elementos que componen un aula virtual surgen de una adaptación del aula tradicional a la que se agregan adelantos tecnológicos accesibles a la mayoría de los usuarios, y en la que se remplazarán factores como la comunicación cara a cara, por otros elementos.

Básicamente el aula virtual debe contener las herramientas que permitan:

- Distribución de la información.
- Intercambio de ideas y experiencias.
- Aplicación y experimentación de lo aprendido,
- Evaluación de los conocimientos
- Seguridad y confiabilidad en el sistema.

#### **2.2.4.4. Características del Aula Virtual [28]**

**Flexible:** Se desea un producto que sea flexible, es decir que pueda ser escalable a futuro, permitiendo la adición de funcionalidades no contempladas en el diseño inicial pero que obedezcan a cambios en el ambiente donde se desenvuelve el proyecto, a características deseables o funcionalidades que expandan la operatividad del sistema.

Por lo tanto, el sistema debe ser lo suficientemente estable y parametrizado de manera que pueda adaptarse fácilmente a los cambios que se requieran.

**Independencia de la plataforma:** Uno de los puntos determinantes en el diseño de la herramienta es la necesidad de independencia con respecto a la plataforma en que esté corriendo. La idea es que, más allá de los requerimientos mínimos de memoria disponible y espacio de disco, los usuarios que la utilicen no necesiten mayores elementos en sus computadores y/o redes para hacerla funcionar totalmente.

**Construcción en base a Estándares:** La herramienta que se va a diseñar debe cumplir con ciertos estándares que existen actualmente y que otras aplicaciones similares los acatan. Existen organizaciones encargadas de crear estándares, para el tipo de aplicaciones de aprendizaje vía Internet existe la IMS (Instructional Management System).

El proyecto IMS es un consorcio abierto que agrupa a aquellos de la industria privada o miembros educativos principalmente universidades, que están desarrollando aplicaciones para el aprendizaje basado en computadora. Miembros de IMS están desarrollando un conjunto de especificaciones de software para facilitar el crecimiento y viabilidad de aprendizaje distribuido en Internet, la organización se está enfocando en el desarrollo de estándares para objetos de enseñanza.

La herramienta que se va a diseñar debe pensarse cumpliendo con estos estándares del IMS, de esta forma se seguiría el patrón o modelo de desarrollo adoptado por otras organizaciones a la hora de desarrollar herramientas del mismo tipo. Esta actividad tiene que ser monitoreada por los desarrolladores, diseñadores y el administrador de la aplicación, desde el primer momento en que se comience a crear la herramienta.

**Acceso, seguridad y configuración:** “Acceso” es una de las palabras claves en todo este contexto. Es muy importante que las personas o instituciones que dispongan de la herramienta puedan configurar los distintos roles de acceso y seguridad de acuerdo a sus

necesidades particulares. El Administrador se encarga de llevar el control de acceso al sistema y de configurarlo para proveer la seguridad requerida.

**Ayuda en Línea:** Se requiere que el sistema provea una ayuda en línea, y que ésta ayuda sea acorde con el contexto en el cual se encuentra el usuario en ese momento y con el tipo de usuario que se esté manejando. La ayuda no debe ser perniciososa, se espera que sea objetiva y discreta, que no cause tedio al usuario y le transmita justo lo que necesita saber. Asimismo se debe proveer una ayuda general para la operatividad general del sistema.

Debe contener:

**Herramientas de Creación de Cursos:** son todas aquellas que permiten la generación, estructuración, actualización y publicación de los contenidos de los cursos. Estas herramientas deben ser manejadas por el Docente.

**Herramientas de Administración de Cursos:** son las que ayudan y facilitan una administración eficiente y efectiva de los cursos por parte del Administrador. Entre las funciones de estas herramientas encontramos la creación y mantenimiento de cuentas de usuarios, la categorización así como la actualización y publicación de contenidos.

**Sistemas de Recuperación de Contenido:** es el que ayuda a buscar algún tipo de contenido referido a un curso en un repositorio de datos mediante palabras clave u otro criterio de búsqueda. El Administrador es el que se encarga de mantener al día este sistema, debe haber herramientas automatizadas que mantengan un índice del curso (sílabo). El Alumno utiliza el sistema cuando necesita algún material de aprendizaje, el Docente también usa este sistema para sus actividades y para actualizar el contenido de los cursos.

**Proveer mecanismos automáticos para la publicación y actualización de contenidos:** La web presenta el ambiente propicio para la publicación actualizada y

dinámica de contenidos ya que éste puede ser modificado directamente sobre el medio y los cambios se visualizan inmediatamente.

Los materiales educativos que se pueden proveer a través del web son inmensamente variados. Se deben explotar los medios provistos (imágenes, animaciones, video, audio, etc.).

Un ejemplo de publicación automatizada se refiere a la creación de publicaciones pre-programadas para ser activadas o desactivadas en cualquier momento en particular. Por ejemplo, un docente puede definir una tarea que sea distribuida un día en particular y cuya solución sólo podrá ser entregada hasta cierta fecha y hora. Los mecanismos automatizados deben permitir que los alumnos envíen la tarea resuelta sólo antes de la fecha y hora previstas.

**Proveer diferentes modos (canales) de comunicación:** Hoy en día los sistemas educacionales vía web, necesitan herramientas tanto asíncronas como síncronas. Las herramientas asíncronas incluyen e-mail, grupos de noticias y librerías de documentos. Herramientas síncronas incluyen chat, pizarras y presentaciones interactivas, todos ellos presentados en un ambiente multimedia.

- **Comunicación Sincrónica:** el emisor y el receptor del mensaje en el proceso de comunicación operan en el mismo marco temporal, es decir, para que se pueda transmitir dicho mensaje es necesario que las dos personas estén presentes en el mismo momento. Estos recursos sincrónicos se hacen verdaderamente necesarios como agente socializador, imprescindible para que el alumno que estudia en la modalidad a distancia no se sienta aislado.

Entre los servicios que dispone Internet en la modalidad sincrónica se pueden mencionar: Videoconferencia y Chat

- **Comunicación Asíncrona:** permiten la transmisión de un mensaje entre el emisor y el receptor sin que tengan que coincidir para interactuar en el mismo instante. Requieren necesariamente de un lugar físico y lógico (como un

servidor, por ejemplo) en donde se guardarán y tendrá también acceso a los datos que forman el mensaje.

Los servicios asincrónicos constituyen los recursos más valiosos para su utilización en la modalidad de educación a distancia, ya que el acceso en forma diferida en el tiempo de la información se hace absolutamente necesario por las características especiales que presentan los alumnos que estudian en esta modalidad (limitación de tiempos, cuestiones familiares y laborales, etc.).

Entre estos servicios se pueden citar los siguientes: Página web, World Wide Web, E-mail, Foros de discusión,

Tabla II. I: Posibles Herramientas

Herramienta	Categoría	Función Común
Presentaciones interactivas	Síncrona	Presentación sincronizada
Chat	Síncrona	Comunicaciones en tiempo real basadas en Texto.
E-mail	Asíncrona	Correo electrónico
Foro de Discusión	Asíncrona	Discusiones
Librería de documentos	Asíncrona	Colocar y repartir documentos

#### 2.2.4.5. Formatos De Presentación

- a) **En línea:** Acceso a Internet mediante la asignación de un usuario y su respectiva clave de acceso
- b) **Local:** Estructura y diseño en ambiente gráfico para su lectura en cualquier navegador, pero con interacción total al grupo de trabajo.

#### 2.2.4.6. Ventajas Del Aula Virtual [13]

- Permite manejar tiempos a conveniencia del estudiante.

- La tecnología como herramienta permite conocer, cuantificar, sistematizar lo que el cliente está solicitando.
- Mejorar las organizaciones como sistemas, a través del desarrollo de programas de calidad total en las dependencias, tener mejores controles, bases de datos históricos, búsqueda avanzada de información, almacenamiento y administración del conocimiento.
- Capacitar a la gente con medios a distancia con costos baratos.
- Supera las limitaciones de tiempo y espacio.
- Desarrolla una amplia cultura computacional.
- Enriquecimiento del aprendizaje.
- Desarrolla un pensamiento creativo y constructivo.
- Se adquiere un criterio más rico y tolerante ante la gran diversidad cultural.
- Ahorro en viajes.
- El usuario establece su propio horario adaptándolo a sus necesidades.
- No precisa de desplazamiento por parte del profesor o alumnado para poder acceder a la educación.
- Permite que el aprendizaje se prolongue durante toda la vida y sea mucho más actualizado.
- El sujeto puede ser autodidacta.
- Permite acceder a la educación desde cualquier lugar del mundo, por lo que permite mejor acceso y más igualdad.
- Desaparece la masificación.
- Al igual que en un Aula tradicional el profesor siempre está disponible.
- El alumno puede seleccionar al profesor que desee, solventando problemas tales como que el alumno se siente incómodo con su profesor y como consecuencia de esto no aprende.

#### **2.2.4.7. Desventajas De Las Aulas Virtuales**

- El ritmo de cambio de la tecnología es muy rápido y los profesores y alumnos no pueden seguir el ritmo de cambio de dicha tecnología.
- El precio de la implementación de esta tecnología es alto.

- La motivación del alumno puede ser complicada.
- Si en la enseñanza presencial ya es complicado poder estimular actitudes emotivas positivas que mejoren el rendimiento académico, en la enseñanza a distancia el problema adquiere dimensiones mayores.
- Se reducen el tipo de relaciones sociales que se establecen en las aulas tradicionales.

### **2.3. SISTEMAS DE ADMINISTRACIÓN DE APRENDIZAJE (LMS)**

Un Ambiente Virtual de Aprendizaje (AVA) o Virtual Learning Environment (VLE) es un sistema de software diseñado para facilitar a profesores la gestión de cursos virtuales para sus estudiantes, especialmente ayudándolos en la administración y desarrollo del curso. Se basan en el principio de aprendizaje colaborativo donde se permite a los estudiantes realizar sus aportes y expresar sus inquietudes en los foros, además van apoyados de herramientas multimediales que hagan más agradable el aprendizaje pasando de ser simplemente un texto en línea, a un entorno interactivo de construcción de conocimiento. [30]

Tales sistemas e-learning a veces también son llamados:

- Sistemas de Gestión del Aprendizaje, del inglés Learning Management System (LMS)
- Sistema de Gestión de Curso (CMS, del inglés Course Management System)
- Sistema de Gestión de Contenido para el Aprendizaje (LCMS, del inglés Learning Content Management System)
- Ambientes de Aprendizaje Gestionado (MLE, del inglés Managed Learning Environment)
- Sistema de Apoyo al Aprendizaje (LSS, del inglés Learning Support System)
- Plataforma de Aprendizaje (LP, del inglés Learning Platform);
- es un medio de educación llamado “comunicación mediante el computador” (CMC, del inglés Computer-Mediated Communication)
- "educación en línea" (OE, del inglés online education)

Una plataforma e-learning, plataforma educativa web o Entorno Virtual de Enseñanza y Aprendizaje es una aplicación web que integra un conjunto de herramientas para la enseñanza-aprendizaje en línea, permitiendo una enseñanza no presencial (e-learning) y/o una enseñanza mixta (b-learning), donde se combina la enseñanza en Internet con experiencias en la clase presencial.

Los Sistemas de Gestión de Aprendizaje son una aplicación software basada en la web que permite planear, implementar, monitorear y principalmente evaluar procesos de aprendizaje específicos. Un sistema de Gestión de aprendizaje tiene instructores que pueden crear o brindar contenidos, monitorear la participación de los estudiantes dentro del sistema, además de evaluar la actuación y desarrollo de los mismos al mismo tiempo, un Sistema de Gestión de Aprendizaje promueve en los estudiantes diferentes habilidades para el uso de herramientas interactivas como foros de discusión, videoconferencias, tele conferencias, chat y correo electrónico. Los sistemas de Gestión de Aprendizaje proporcionan la funcionalidad al e-learning, ya que es el espacio en donde los estudiantes pueden planear, acceder, enviar y gestionar el aprendizaje por sí mismos, interactuar con otros alumnos, con los monitores y facilitadores. (Castro, 2007). [21]

En los LMS los módulos con funciones administrativas permiten, por ejemplo, configurar cursos, matricular alumnos, registrar profesores, asignar cursos a un alumno, llevar informes de progreso y calificaciones. Los LMS también facilitan el aprendizaje distribuido y colaborativo a partir de actividades y contenidos pre-elaborados, de forma síncrona o asíncrona, utilizando los servicios de comunicación de Internet como el correo, los foros, las videoconferencias y el chat.

El alumno interactúa con la plataforma a través de una interfaz web que le permite seguir las lecciones del curso, realizar las actividades programadas, comunicarse con el profesor y con otros alumnos, así como dar seguimiento a su propio progreso con datos estadísticos y calificaciones.

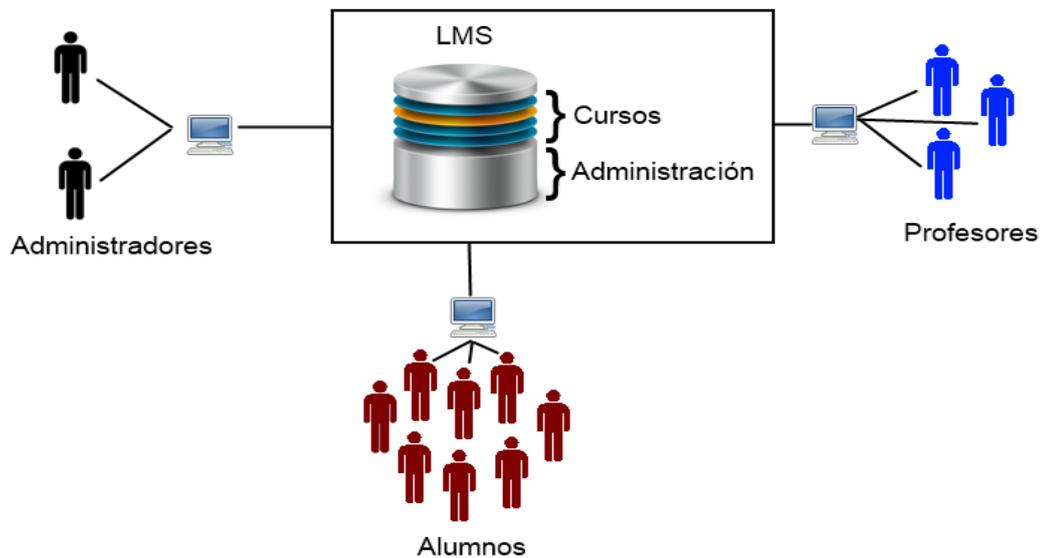


Ilustración II. 1: Sistemas de Gestión de Aprendizaje (LMS)

### 2.3.1. Funciones y Características de un LMS [22]

Para poder cumplir su propósito un Sistema De Gestión De Aprendizaje debe poseer un conjunto mínimo de funciones. Estas pueden agruparse de la siguiente forma:

- **Distribución de contenidos:** editor de contenidos en línea, repositorios de archivos de imágenes, de video y de texto como biblioteca en línea, sistemas de reconocimiento de contenidos en CD, inserción de hipervínculos, imágenes y videos y administración de calendario de contenidos.
- **Comunicación y Colaboración:** foros de discusión por curso, sala de chat por curso, formación de grupos de trabajo, comunicación con el tutor, miembros del curso, novedades y calendario del curso.
- **Seguimiento y evaluación:** estadísticas y ficha personal por alumno, seguimiento de cada actividad, sistemas de exámenes editables por el docente o tutor y reportes de actividad.
- **Administración y asignación de permisos:** otorgamiento de permisos y actualizaciones, asignación de permisos por perfil de usuario, administración personal de perfiles de usuario, proceso de inscripción, planes de carrera y oferta formativa.

### 2.3.2. Beneficios de los LMS [4]

- **Siempre disponible:** Accesible las 24 horas en los 7 días desde cualquier lugar con acceso a Internet. Varios usuarios pueden entrar al sistema en cualquier momento.
- **Educación centralizada:** En lugar de buscar en varios sistemas y archivos de papel para averiguar o encontrar algo, todo estará disponible en un sistema único.
- **Simplificación de los procesos de aprendizaje:** Asegura que cada empleado reciba la formación adecuada así como ayuda a administrar mejor los salones, instructores, equipos y otros recursos necesarios para impartir los cursos.
- **Reducir los costos de capacitación:** incluyendo viajes (transporte), alojamiento, instalaciones materiales, enseñanza y otros.
- **Aumento en la velocidad de aprendizaje:** Entre más rápido el empleado aprenda y se capacite, mejor para la empresa.
- **Ayuda a cumplir con las regulaciones:** El sistema proporciona reportes y pruebas para cumplir con los requisitos legales. Hay industrias (farmacéutica, construcción o de petróleo) que requieren estos para evaluar e informar sobre los efectos del cumplimiento.
- **Interactividad:** Permite aportar documentación electrónica o enlaces al instante.
- **Participativo:** El capacitador se puede comunicar fácilmente con todos los asistentes y no solamente con uno en particular. Además, estimula la formulación de preguntas a través de foros u otras actividades.

### Beneficios para Estudiantes [5]

- **Capacitación centrada en el estudiante:** brindándole las herramientas y el entorno adecuado para que aprenda a su propio ritmo, con la mezcla de elementos que se ajuste a su estilo de aprendizaje.
- **Flexibilidad:** capacitándose en cualquier lugar, a cualquier hora, y respondiendo a las necesidades cambiantes que su puesto de trabajo le exija.

- **Personalización:** proporcionando al estudiante la combinación correcta de cursos y contenidos de acuerdo a sus conocimientos y requerimientos.
- **Soporte:** el estudiante puede, mediante herramientas de colaboración, acceder a soporte sincrónico o asincrónico, de un profesor, experto en contenido o un facilitador que brinde asistencia durante las capacitaciones.
- **Modalidades:** el estudiante tendrá acceso a eventos sincrónicos (a la misma hora y en cualquier lugar) o eventos asincrónicos (a cualquier hora y en cualquier lugar) dependiendo de su interés y tipo de recurso disponible.

### **Beneficios del administrador**

- **Administración comprensiva de capacitación:** seguimiento detallado del avance de cada estudiante, resultados de cursos, asignación de recursos (desde reuniones virtuales hasta la reservación de salones de clase tradicionales) y alineamiento de metas corporativas con iniciativas de capacitación.
- **Rutas de aprendizaje y competencias:** Los procesos de asignación de cursos a estudiantes o grupos de estudiantes se simplifica notablemente mediante el desarrollo de rutas de aprendizaje predeterminadas o el desarrollo de competencias para cada puesto de trabajo, brindando la oportunidad de que el sistema y/o el estudiante escojan la capacitación correcta para su puesto de trabajo.
- **Velocidad de adopción de conocimiento:** una variable cada vez más crítica para las empresas. Los LMS permiten llevar grandes cantidades de conocimiento a una audiencia extensa en muy poco tiempo y con pocos recursos, logrando resultados en la operación del negocio a una velocidad sin precedentes.

### **2.3.3. Herramientas [17]**

Los sistemas de gestión del aprendizaje, LMS, aportan importantes herramientas al proceso educativo, generalmente:

- **Herramientas de gestión y distribución de contenidos:** Permiten almacenar, organizar, recuperar y distribuir contenidos educativos estructurados en contenidos educativos y estructurarlos en contenidos de mayor complejidad y alcance temático.
- **Herramientas de administración de usuarios:** Facilitan el registro de los usuarios del sistema para el posterior control de acceso y presentación personalizada de los contenidos y cursos.
- **Herramientas de comunicación:** Chats, foros, correo electrónico, tableros de anuncios, permiten la comunicación entre estudiantes y tutores en una vía o en doble vía sincrónica y asincrónicamente.
- **Herramientas de evaluación y seguimiento:** Apoyan la construcción y presentación de evaluaciones mediante la utilización de diferentes tipos de preguntas: abierta, falso o verdadero, selección múltiple, múltiple opción, completar y apareamiento entre otras. Algunas veces también permite la construcción de bancos de preguntas usados con frecuencia para seleccionar aleatoriamente preguntas para los estudiantes.

#### 2.3.4. Evolución de los Sistemas de Gestión de Aprendizaje [6]

Inicialmente fueron diseñados para facilitar a los administradores y profesores la gestión y organización administrativa de curso virtuales, relegando los aspectos pedagógicos a un segundo plano, principalmente por desconocimiento sobre el rol que las tecnologías podían desempeñar en los procesos educativos de este modo, los LMS pasaron a convertirse en prolongaciones, clones o extensiones lógicas de los espacios de formación presenciales.

El origen de las plataformas de e-learning, basadas en web, se debe a una especialización de los CMS (Sistemas de Gestión de Contenidos), en sistemas orientados a la gestión de contenidos para el aprendizaje a distancia.

**CMS:** Los Sistemas de Gestión de Contenidos (Content Management System) es un software que se utiliza principalmente para facilitar la gestión de webs, ya sea en

Internet o en una intranet, y por esto también son conocidos como Gestores de Contenido Web (Web Content Management o WCM).

Cabe tener en cuenta, sin embargo, que la aplicación de los CMS no se limita solo a las webs, y en el caso del *e-learning* la gestión no está centrada en la web, sino en los contenidos educativos (recursos, documentos y pruebas evaluadoras, RLO (Reusable Learning Objects) entre otros).

Los CMS en los últimos años han progresado en tres etapas evolutivas, que han impactado, cada vez de forma más notoria, sobre la velocidad de creación de contenidos, el coste, la flexibilidad, la personalización del aprendizaje, la calidad en la atención del estudiante y las ventajas competitivas de las organizaciones que han aplicado las soluciones de *e-learning*.

- **Primera etapa:** Los CMS (Content Management System o Course Management System) son dentro de las plataformas de *e-learning* los más básicos y permiten la generación de sitios web dinámicos. El objetivo de estos programas es la creación y gestión de información en línea (textos, imágenes, gráficos, videos, sonido, etc.). También se caracterizan por no poseer herramientas elaboradas de colaboración (foros, chats, diarios, etc.) ni apoyo en tiempo real.
  
- **Segunda etapa:** los LMS (Learning Management System) aparecen a partir de los CMS y proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación de la web con la colaboración de múltiples usuarios. Están orientados al aprendizaje y la educación, proporcionando herramientas para la gestión de contenidos académicos, permitiendo mejorar las competencias de los usuarios de los cursos y su intercomunicación, en un entorno donde es posible adaptar la formación a los requisitos de la empresa y al propio desarrollo profesional. Disponen de herramientas que permiten la distribución de cursos, recursos, noticias y contenidos relacionados con la formación general.

- **Tercera etapa:** los LCMS (Learning Content Management System) son plataformas que integran las funcionalidades de los CMS y los LMS, que incorporan la gestión de contenidos para personalizar los recursos de cada estudiante y donde las empresas se convierten en su propia entidad editora, con autosuficiencia en la publicación del contenido de una forma sencilla, rápida y eficiente, resolviendo los inconvenientes de las anteriores plataformas. Ofrecen facilidad en la generación de los materiales, flexibilidad, adaptabilidad a los cambios, control del aprendizaje y un mantenimiento actualizado del conocimiento.

Los LCMS añaden técnicas de gestión de conocimiento al modelo de los LMS en ambientes estructurados y diseñados para que las organizaciones puedan implementar mejor sus procesos y prácticas, con el apoyo de cursos, materiales y contenidos en línea. Permiten una creación muy eficiente por parte de sus desarrolladores, expertos colaboradores o instructores que participan en la creación de contenidos.

### **2.3.5. Plataformas de Enseñanza Virtual Actuales [26]**

En la actualidad existe una amplia oferta de LMS los cuales podríamos dividir en dos grupos fundamentales, los de Software Libre y los de Software Privado. Los LMS de Software Libre permiten ser usados sin necesidad de un coste en la compra del software ni pagos por licencias. Por este motivo este tipo de plataformas de enseñanza se ajustan más a los intereses de las universidades públicas y serán analizados con detenimiento en este documento.

A continuación podemos observar un listado con las principales plataformas de enseñanza virtual organizadas de acuerdo al tipo de software, que son:

Tabla II. II: Principales LMS

LMS Software Libre	LMS Software Privado
Moodle	ECollege
Sakai	EDoceo
Claroline	Desire2Learn
Docebo	Blackboard
Dokeos	Skillfactory
Ilias	Delfos LMS
LRN	Prometeo
ATutor	Composica
Lon-CAPA	WebCT

Todas las plataformas aquí expuestas cumplen con las funcionalidades comentadas anteriormente y muchas de ellas son usadas en la actualidad por las universidades para implementar sus campus virtuales.

### 2.3.6. Plataformas de enseñanza virtual libres

Las universidades intentan modernizar las metodologías de enseñanza tradicional con el uso de sistemas E-Learning y crear con ello los llamados campus virtuales. Por otro lado, debido a la filosofía de las universidades de preservar y compartir el conocimiento y a la necesidad de optimizar los costes, el software libre se perfila como el candidato perfecto para que las universidades materialicen estos objetivos.

En este apartado se explicará el concepto de software libre y las distintas licencias englobadas bajo este término genérico. Esto es fundamental, puesto que las licencias limitan qué podremos hacer con el Software y qué licencia tendrá nuestro Software si deriva de otro.

### 2.3.6.1. Software Libre [34]

Como se ha comentado anteriormente, existe una modalidad de Software conocida como Software Libre, pero este concepto es muy genérico y contiene un gran número de matices.

Lo primero es clarificar el concepto de Software “Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.” (IEEE 1990) Considerando esta definición, el concepto de software va más allá de los programas de cómputo en sus distintos estados: código fuente, binario o ejecutable; también su documentación, datos a procesar e información de usuario forman parte del software. Simplificando la definición para nuestro ámbito diremos que son las instrucciones y datos necesarios para que un computador realice una operación y además el ser humano sea capaz de comprender el funcionamiento de dicha operación.

El segundo término que debemos clarificar es Libre, la Free Software Foundation (FSF) expone que: “software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software”. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- **Libertad 0:** La libertad de usar el programa, con cualquier propósito
- **Libertad 1:** La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. El acceso al código fuente es una condición previa para esto.
- **Libertad 2:** La libertad de distribuir copias, con lo que puedes ayudar al prójimo.
- **Libertad 3:** La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Para poder cumplir con estas libertades y con el concepto global de Software, el Software Libre debe distribuirse incluyendo no solo el programa ejecutable por el

computador sino que debe incluir también el código fuente y aconsejablemente la documentación del programa, de esta forma se cumplirá con la libertad número 1.

Una variante más reciente plantea que el Software ha de ser de Código Abierto, este movimiento se centra más en los beneficios prácticos y menos en los ideales éticos que defiende el Software Libre, la principal organización defensora del Código Abierto es la Open Source Initiative (OSI). [24]

OSI sugiere que los siguientes puntos han de caracterizar el Código Abierto:

- **Libre distribución:** No se puede impedir la venta o distribución del programa o parte de él. Así mismo, tampoco se puede exigir el pago de un canon o tasa a cambio de su distribución por parte de terceros.
- **Código fuente:** El programa debe incluir su código fuente y no se puede restringir su redistribución.
- **Trabajos derivados:** No debe impedirse realizar modificaciones o trabajos derivados del programa y debe permitirse que éstos sean distribuidos bajo mismos términos del software original.
- **Integridad del código de fuente original:** Puede exigirse que una versión modificada del programa tenga un nombre y número de versión diferente que el programa original para poder proteger al autor original de la responsabilidad de estas versiones.
- **No discriminación de personas o grupos:** Las condiciones de uso del programa no pueden discriminar a una persona o a un grupo de personas.
- **No discriminación de uso:** No se puede negar a ninguna persona hacer uso del programa para ningún fin como, por ejemplo, comercial o militar.
- **Distribución de la licencia:** Los derechos del programa deben aplicarse a todos los que redistribuyen el programa sin ninguna condición adicional.
- **La licencia no debe ser específica de un producto:** Los derechos garantizados al usuario del programa no deben depender de que el programa forme parte de una distribución o paquete particular de software.

- **La licencia no debe restringir otro software:** La licencia no debe poner restricciones a otros programas que se distribuyen junto con el software licenciado.
- **La licencia debe ser tecnológicamente neutra:** No puede existir ninguna disposición de la licencia que obligue al uso de una tecnología concreta. A continuación vamos a proporcionar una breve explicación sobre algunas de las licencias de Software Libres más importantes.

### 2.3.6.2. Licencias de Software Libre [25]

Una licencia es un contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquél a quien éste haya cedido los derechos de explotación, quién elige la licencia según la cual distribuye el software (OSI, 2005).

- **Licencias GLP:** Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autor (copyright) y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL.

La licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia. Y añade que si se reutiliza en un mismo programa código "A" licenciado bajo licencia GNU GPL y código "B" licenciado bajo otro tipo de licencia libre, el código final "C", independientemente de la cantidad y calidad de cada uno de los códigos "A" y "B", debe estar bajo la licencia GNU GPL.

En la práctica, esto hace que las licencias de software libre se dividan en dos grandes grupos: aquellas que pueden ser mezcladas con código licenciado bajo

GNU GPL (y que inevitablemente desaparecerán en el proceso, al ser el código resultante licenciado bajo GNU GPL) y las que no lo permiten.

- **Licencias AGLP:** La Licencia Pública General de Affero (en inglés Affero General Public License, también Affero GPL o AGPL) es una licencia copy left derivada de GNU GPL diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.
  
- **Otras Licencias:** Como hemos comentado el número de licencias disponibles es casi infinito. Conviene saber dónde buscar las condiciones de las licencias disponibles, para ello se pueden usar dos páginas:
  - Open Source Licenses de OSI: donde podemos encontrar una lista de licencias organizadas por categorías que contiene las licencias aprobadas por OSI. WEB
  - License List de GNU: donde encontramos un listado de licencias sus características y la compatibilidad con la licencia GNU GPL. WEB

#### **2.4. ANÁLISIS DE LAS PLATAFORMAS PARA EDUCACIÓN VIRTUAL MÁS REPRESENTATIVAS**

Tanto los LMS (Learning Management System) como los LCMS (Learning Content Management System), que veremos en el apartado siguiente, se pueden generalizar como Sistemas de Gestión de Aprendizaje, ya que el primero gestiona la parte administrativa de los cursos así como el seguimiento de actividades y avance del alumno. El segundo gestiona el desarrollo de contenidos, su acceso y almacenamiento. En el entorno educativo, los más comunes son los LMS ya que la complejidad de los LCMS los ha llevado a un desarrollo más lento. Debemos tener en cuenta que una determinada herramienta (LMS o LCMS) no está directamente ligada a un paradigma educativo concreto (e-learning, b-learning), sino que cualquier sistema de gestión de aprendizaje puede adaptarse a las necesidades educativas concretas de cada institución.

Para la selección de los dos LMS más adecuados para la agregación de componentes, se realizó una comparativa entre 5 de los múltiples LMS de software libre existentes para esto se ha tomado como referencia en la investigación **L.M.S. Y WEB 2.0 APRENDER Y COLABORAR POR INTERNET C.U.N. 2011** [11] desarrollada por Jorge David Castillo Espejo de la Corporación Unificada Nacional de Educación Superior CUN. Y la investigación **PLATAFORMAS DE ENSEÑANZA VIRTUAL LIBRES Y SUS CARACTERÍSTICAS DE EXTENSIÓN: DESARROLLO DE UN BLOQUE PARA LA GESTIÓN DE TUTORÍAS EN MOODLE** [17] desarrollada por Diego Macías Álvarez de la Universidad de ALCALA. Estas investigaciones se han seleccionado como guía principal para seleccionar las herramientas LMS más populares.

Y es así que se ha considerado como herramientas LMS más comunes las siguientes: Moodle, Dokeos, ATutor, Claroline y Sakai. Y basándonos en las sus características se ha dividido en 2 parámetros cada uno con sus sub-parámetros para determinar por separado sus potencialidades y debilidades. Se describe a continuación los parámetros de comparación para la determinación de los LMS más aptos para la agregación de componentes.

#### **2.4.1. Descripción de los parámetros de comparación**

##### **2.4.1.1. Madurez del producto**

Tabla II. III: Descripción del Parámetro 1- Madurez del Producto

		<b>Descripción</b>
<b>Parámetro 1</b>	<b>Madurez del producto</b>	Describe la madurez del producto como herramienta LMS de una manera sencilla.
[Indicador 1.1]	Idiomas	Describe la cantidad de lenguajes que soporta cada herramienta LMS.
[Indicador 1.2]	Plataforma	Describe la o las plataformas que soportan cada una de las herramientas

		<b>Descripción</b>
		LMS.
[Indicador 1.3]	Manejador de Base de datos	Describe la o las Base de datos que porta las herramientas.
[Indicador 1.4]	Servidor Web	Describe el Servidor en el que se ejecutará el LMS.
[Indicador 1.5]	Lenguaje de programación	Describe los lenguajes de programación en los que son desarrollados los LMS.

#### 2.4.1.2. Personalización

Tabla II. IV: Descripción del Parámetro 2- Personalización

		<b>Descripción</b>
<b>Parámetro 1</b>	<b>Personalización</b>	Describe capacidad de personalización o no, de las herramientas LMS antes mencionadas.
[Indicador 2.1]	Personalización de la Apariencia	Establece si es posible la personalización de la apariencia (temas) de los diferentes LMS.
[Indicador 2.2]	Niveles de Acceso	Describe los diferentes niveles de acceso que poseen los LMS.
[Indicador 2.3]	Estándares	Describe que estándares cumplen los LMS para su funcionamiento y funcionalidades.
[Indicador 2.4]	Documentación	Describe la documentación existente acerca de los LMS para desarrolladores.
[Indicador 2.5]	Personalización y agregación de nuevos componentes	Describe si el LMS permite o no la personalización de sus componentes así como la creación de nuevos

		Descripción
		componentes.

## 2.4.2. Descripción de las Plataformas de educación virtual

A continuación se detallará las plataformas a estudiar y posteriormente se realizará una comparativa de las 5 herramientas para proceder a seleccionar las dos plataformas más adecuadas en la agregación de componentes.

### 2.4.2.1. Descripción de la plataforma Moodle [23]



Ilustración II. 2: Logotipo Moodle

La palabra Moodle es el acrónimo de Modular Object-Oriented Dynamic Learning Environment (Entorno Modular de Aprendizaje Dinámico Orientado a Objetos).

Moodle fue creado por Martin Dougiamas, un administrador de WebCT en Curtin University, Australia, y graduado en Ciencias de la Computación y Educación.

Moodle posee una interfaz intuitiva, ligera y compatible que facilita a los diseñadores de cursos la construcción de foros, chats, lista de correo, glosarios, bibliotecas, autoevaluaciones y demás funcionalidades. Los recursos disponibles en Moodle se pueden clasificar en tres categorías: Recursos Transmisivos, Recursos Interactivos y Recursos Colaborativos. A estas tres categorías también podemos añadir las herramientas de comunicación.



Ilustración II. 3: Interfaz Moodle

Moodle tiene una base numerosa de usuarios: hay 67 000 sitios registrados, que ofrecen 5,5 millones de cursos, en los que participan 54 millones de usuarios. La plataforma está traducida a 86 idiomas. [14]

Cada uno de los recursos que componen la plataforma se define como una pieza de software independiente que puede ser modificada o eliminada según las necesidades de la institución de enseñanza que la adopte. De la misma forma también es posible instalar nuevos módulos que añadan funcionalidad al sistema, bien descargándolos desde la página oficial de módulos no estándar de Moodle o bien desarrollándolos a medida. El proceso de desarrollo de un módulo no es complejo, pero requiere conocer un poco más a fondo la arquitectura y el funcionamiento interno del sistema.

### Características de Moodle

- **Gran disponibilidad:** Permite que los contenidos, estén disponibles en cualquier momento, satisfaciendo las necesidades de los agentes que intervienen en la creación, adecuación, administración y recepción de material formativo.
- **Escalabilidad:** La herramienta está en capacidad de adaptarse en marcha a los requerimientos que vayan surgiendo como producto de la utilización de la misma.
- **Facilidad de uso:** Existe muy buena documentación que facilita su utilización.

- **Interoperabilidad:** Utiliza estándares abiertos de la industria (XML, SOAP, etc.) y se puede ejecutar en diferentes sistemas operativos.
- **Estabilidad:** Posee un entorno eficaz y confiable.
- **Seguridad:** La implementación de perfiles de usuario con funciones claramente definidas entre sí, la restricción de acceso a comunidades de aprendizaje y la implementación de uso de contraseñas para acceder a un determinado contenido, colaboran para que el sistema evite riesgos innecesarios.

Moodle se distribuye gratuitamente como Software libre (Open Source) (bajo la Licencia Pública GPL de GNU). Básicamente esto significa que Moodle tiene derechos de autor (copyright), pero que se tiene algunas libertades. Puede copiar, usar y modificar Moodle siempre que acepte: proporcionar el código fuente a otros, no modificar o eliminar la licencia original y los derechos de autor, y aplicar esta misma licencia a cualquier trabajo derivado de él.

Promueve una pedagogía constructivista social (colaboración, actividades, reflexión crítica, etc.). Su arquitectura y herramientas son apropiadas para clases en línea, así como también para complementar el aprendizaje presencial. [1]

La instalación es sencilla requiriendo una plataforma que soporte PHP y la disponibilidad de una base de datos. Moodle tiene una capa de abstracción de bases de datos por lo que soporta los principales sistemas gestores de bases de datos entre las que se incluyen: MySQL, PostgreSQL, Microsoft SQL y Oracle (MySQL de preferencia).

Se ejecuta sin necesidad de cambios en el sistema operativo bajo Unix, Linux, Windows, Mac OS X, Netware y todos aquellos sistemas operativos que permitan PHP. Moodle es un ejemplo de aplicación LAMP (Linux, Apache, MySQL y PHP). Se trata fundamentalmente de aplicaciones web escritas con un lenguaje de script, que utilizan una base de datos SQL para almacenar información. Con el incremento de la popularidad de ejecutar aplicaciones web de código abierto en Windows y en Mac OS X, han surgido dos nuevos términos: WAMP y MAMP.

Normalmente trabajará con cualquier servidor web que soporte la versión apropiada de PHP. En la práctica, el servidor web más usado es Apache, que está disponible para la mayoría de sistemas operativos. Internet Information Services (IIS) es otro servidor web muy popular para alojar Moodle.

Moodle cuenta con los últimos recursos en e-learning y está adaptada a los estándares más importantes, como son AICC y el SCORM, para la integración de los contenidos de formación con la plataforma on-line.

Además, dispone de los recursos más frecuentes como chat, foros de debate, mensajería, tablón de anuncios, sistema de seguimiento de accesos, utilidades para la creación de actividades cooperativas, etc.

Al ser una plataforma tan abierta, Moodle debe ser riguroso con su seguridad. Por ello hace grandes esfuerzos para asegurar que ninguna persona acceda al sistema si no debe hacerlo, y que los usuarios que acceden lo hagan de la manera correcta.

Los permisos en Moodle pueden ser asignados dentro de seis contextos: sitio/global, categoría de curso, curso, bloques y actividades, usuario y portada. Como desarrolladores, podemos crear capacidades para controlar el acceso a las nuevas funcionalidades que despleguemos en el sistema, y también se puede crear cualquier número de roles personalizados a partir de la lista de unas doscientas capacidades del sistema.

De la misma forma también es posible instalar nuevos componentes que añadan funcionalidad al sistema, bien descargándolos desde la página oficial de módulos no estándar de Moodle o bien desarrollándolos. El proceso de desarrollo de un módulo o componente no es complejo, pero requiere conocer un poco más a fondo la arquitectura y el funcionamiento interno del sistema.

Los roles estándar del sistema son:

- **Administrador:** El administrador del sistema tiene todos los permisos.
- **Creador de cursos:** Puede crear cursos en el sistema y puede estar limitado a una categoría de cursos.
- **Profesor:** Puede administrar un curso, y además puede desarrollar y actualizar su contenido.
- **Profesor no editor:** Puede administrar un curso, pero no puede modificar su estructura.
- **Estudiante:** Puede estar matriculado en un curso.
- **Usuario autenticado:** Todo usuario que haya iniciado sesión en el sistema tiene este rol.
- **Invitado:** Los usuarios no autenticados que tiene permiso de acceso al sistema.

#### 2.4.2.2. Descripción de la plataforma ATutor [2]



Ilustración II. 4: Logotipo ATutor

ATutor es un Sistema de Gestión de Contenidos de Aprendizaje (LCMS) de Código abierto basado en la Web y diseñado con el objetivo de lograr accesibilidad y adaptabilidad. Los administradores pueden instalar o actualizar ATutor en minutos.

Este proyecto empezó en 2002 en colaboración con el Adaptive Technology Resource Centre (ATRC) de la Toronto University. Este centro es un líder internacionalmente reconocido en el desarrollo de tecnologías y estándares que permitan a la gente con discapacidades el acceso a las oportunidades e-learning y esta misión ha influenciado profundamente el desarrollo de la plataforma.

El software se cita como único por sus características de accesibilidad, (útil para los alumnos con discapacidad visual u otras discapacidades); y por su idoneidad para el uso educativo de acuerdo a los criterios de evaluación del software creado por la Sociedad Americana para Capacitación y Desarrollo (ASTD).

Los educadores pueden rápidamente ensamblar, empaquetar y redistribuir contenido educativo, y llevar a cabo sus clases online. Los estudiantes pueden aprender en un

entorno de aprendizaje adaptativo. ATutor es un programa diseñado en PHP, Servidor Web Apache y Base de Datos MySQL, trabaja sobre plataformas Windows, GNU/Linux, Unix, Solaris, soporte a 32 idiomas, contiene herramienta de Gerencia y administra alumnos, tutores, cursos y evaluaciones en línea, herramienta de Autoría incorporada, herramienta de Colaboración incorporada. La incorporación de las especificaciones de empaquetado de contenido IMS/SCORM, permiten que los diseñadores de contenidos creen contenido reutilizable que se puede intercambiar entre diversos sistemas de aprendizaje. [31]

ATutor es el LCMS primero en cumplir completamente con las especificaciones de accesibilidad de W3C WCAG 1.0 en la AA + nivel, que permite el acceso a todos los contenidos incluidos del sistema en todos los niveles de usuario con privilegios, incluyendo las cuentas de administrador. Su conformidad con XHTML 1.0 está diseñado para asegurar que ATutor se presenta y que aparecen constantemente en cualquier tecnología compatible.

El Navegador Web: ATutor puede ser soportado por muchos navegadores Web, se recomienda los navegadores como Firefox y Microsoft Internet Explorer.

### **Características de ATutor**

- Accesibilidad
- Diseño simple y robusto
- Tiene gestor de contenidos accesible y herramientas para validar los contenidos que se generan.
- Presenta un sistema modular para presentación y código.
- Es el único LCMS de software libre.
- Tiene herramientas diferenciadas para los administradores, profesores y estudiantes.

ATutor presenta 3 clases de perfiles de usuarios:

- **Administrador:** Puede tener múltiples administradores, Gestiona las inscripciones de los usuarios, Maneja los cursos, Muestra listas de estudiantes, puede personalizar el ambiente educativo, Se encarga del mantenimiento de la plataforma (actualizaciones, idioma, seguridad, etc.)
- **Instructor:** Funcionalidades, Inscripción y control de acceso a los estudiantes, Administra los contenidos, Administra estudiantes y monitores, Exporta contenido.
- **Estudiante:** Están disponibles herramientas para garantizar accesibilidad a los contenidos, Poseen sistema de mensajería y herramientas para trabajar en equipo, Disponen de glosarios, foros, wikis, listas de lecturas y buscador.

### 2.4.2.3. Descripción de la Plataforma Dokeos [12]



Ilustración II. 5:  
Logotipo Dokeos

**Dokeos** es un entorno de e-learning y una aplicación de administración de contenidos de cursos y también una herramienta de colaboración. Es software libre y está bajo la licencia GNU GPL, el desarrollo es internacional y colaborativo. También está certificado por la OSI y puede ser usado como un sistema de gestión de contenido (CMS) para educación y educadores. Esta característica para administrar contenidos incluye distribución de contenidos, calendario, proceso de entrenamiento, chat en texto, audio y video, administración de pruebas y guardado de registros. Hasta el 2007, estaba traducido en 34 idiomas (y varios están completos) y es usado por más de mil organizaciones.

Los permisos son implementados en base a los roles. Existen 4 roles predefinidos: administrador, profesor, usuario, invitado. No es posible modificar estos roles o crear un nuevo rol. Los usuarios pueden ser organizados dentro de grupos. Una de las principales diferencias con Claroline es que antes de proceder a la instalación de Dokeos debemos crear una base de datos en nuestro servidor MySQL de forma manual.

Además no es posible dividir la aplicación en varias bases de datos. Este hecho implica que el diagrama de despliegue de Dokeos sea más simple, aunque también menos flexible y potente.

Dokeos se basa en la especificación SCORM para el manejo de contenidos. Puede ser utilizada como acompañamiento a las clases presenciales de los estudiantes, para educación totalmente virtual y como ayuda en instituciones semi-presenciales.

No se requiere de entrenamiento técnico específico en la plataforma para poder utilizarla, presenta un entorno fácil de aprender y adaptable a las necesidades de las instituciones y organizaciones. Gracias a la implementación de la especificación SCORM, Dokeos proporciona mecanismos para la gestión del aprendizaje, porque permite importar, exportar, generar test sofisticados, etc., También permite mostrar presentaciones en PowerPoint, presenta informes detallados que se pueden exportar a Excel, Business Objects y tiene una herramienta de videoconferencia para sesiones de aprendizaje en línea.

Para la puesta en marcha de la plataforma Dokeos se requiere un Servidor Web HTTP (Recomiendan Apache), con Base de Datos MySQL y desarrollado en lenguaje PHP.

Dokeos posee herramientas de enseñanza y seguimiento como: foros, chat, blogs, videoconferencia, intercambio de documentos y agenda. Por otro lado posee herramientas para administración como: importación de usuarios, gestión de las sesiones formativas, es compatible con OpenID y LDAP.

#### **2.4.2.4. Descripción de la Plataforma Claroline [9]**



Ilustración II. 6: Logotipo Claroline

Claroline es un proyecto de software libre que se distribuye con licencia GNU/GPL. Está escrito en el lenguaje de programación PHP, utiliza como SGBD, MySQL. Sigue las especificaciones de SCORM e IMS. Está disponible para plataformas (Linux) y navegadores libres (Mozilla, Netscape), y plataformas (Unix, Mac OS X y Windows) y navegadores propietarios (Internet Explorer).

Presenta las características propias de un sistema de gestión de contenidos (CMS). Puede ser utilizado por formadores, para administrar cursos virtuales en entornos e-learning.

El proyecto Claroline fue iniciado en el año 2000, en el Instituto Pedagógico Universitario de Multimedia de la Universidad Católica de Lovain (Bélgica), por Thomas De Praetere, Hugues Peeters y Christophe Gesché, con la financiación de la Fundación Louvain de la misma Universidad. Desde 2004, el Centro de Investigación y Desarrollo (CERDECAM), del Instituto Superior de Ingeniería Belga (ECAM), participa en el desarrollo de Claroline, con un equipo financiado por la Región Valona. [32]

Claroline está traducido a 35 idiomas y tiene una gran comunidad de desarrolladores y usuarios en todo el mundo.

Claroline se basa en principios pedagógicos provenientes de la literatura sobre el valor añadido de las tecnologías de training. Desde el año 2000, la principal preocupación de los desarrolladores no es construir un gran número de nuevas características, sino a concentrarse en unas pocas herramientas elaboradas sobre el enfoque pedagógico y la interfaz que ofrece a los usuarios. Además, una gran comunidad mundial de usuarios y desarrolladores de Claroline contribuye al desarrollo y la difusión amplia.

#### **2.4.2.5. Descripción de la Plataforma Sakai [27]**



Ilustración II. 7: Logotipo Sakai

El Proyecto Sakai está desarrollando software educativo de código abierto. El nombre Sakai proviene del cocinero Hiroyuki Sakai. El Proyecto Sakai tiene su origen en la Universidad de Michigan y en la Universidad de Indiana, a las que se unieron el MIT y Stanford University, junto a la Iniciativa de Conocimiento Abierto (OKI) y el consorcio uPortal. El Proyecto se consolidó con generosa ayuda de la Fundación Mellon.

El objetivo del Proyecto Sakai es crear un entorno de colaboración y aprendizaje para la educación superior, que pueda competir con sus equivalentes comerciales Blackboard/WebCT y que mejore otras iniciativas de Código Abierto como Moodle. [35]

Es una comunidad de instituciones académicas, organizaciones comerciales e individuos que trabajan juntos para desarrollar una colaboración común y ambiente de aprendizaje (CLE). El CLE Sakai es una fuente libre, la comunidad, la plataforma de software educativo se distribuye bajo la Licencia para la Educación de la Comunidad (un tipo de licencia de código abierto).

El CLE Sakai se utiliza para la enseñanza, la investigación y la colaboración. Sistemas de este tipo también se conocen como de gestión de cursos (CMS), Learning Management Systems (LMS) o Entornos Virtuales de Aprendizaje (EVA). Sakai es una basada en Java, suite de aplicaciones orientadas a servicios que está diseñado para ser escalable, fiable, interoperable y extensible. La versión 1.0 fue lanzado en marzo de 2005.

El software Sakai incluye muchas de las características comunes a los sistemas de gestión de cursos, incluyendo la distribución de documentos, un libro de calificaciones, la discusión, chat en vivo, asignación de cargas, y las pruebas en línea.

Además de las características de gestión de cursos, Sakai es concebido como una herramienta de colaboración para proyectos de investigación y de grupo. Para apoyar esta función, Sakai incluye la capacidad de cambiar la configuración de todas las herramientas basadas en roles, cambiando lo que el sistema permite a los usuarios diferentes que ver con cada herramienta. También incluye un wiki, lista de distribución de correo y archivo. Los ejemplos pueden incluir sitios para proyectos de colaboración, la enseñanza y carteras.

### 2.4.3. Análisis Comparativo.

Los resultados de los parámetros con sus respectivos indicadores se detallan en un cuadro comparativo de los LMS Moodle, Dokeos, Claroline, Sakai y ATutor.

La calificación para cada parámetro se determinara de acuerdo a la escala que se mostrara a continuación, lo cual nos permitirá determinar los dos LMS más aptos para la agregación de componentes.

Tabla II. V: Valoración cualitativa y cuantitativa.

Regular	Bueno	Muy Bueno	Excelente
<70%	>=70% y <80%	>=80% y <95%	>=95%

La evaluación para los indicadores es de acuerdo a las características estudiadas de los diferentes LMS, para lo cual la valoración variará entre uno y cuatro.

Tabla II. VI: Escala de valoración cualitativa y cuantitativa para los indicadores

Valor Cualitativo		Valor Cuantitativo	Valor Representativo
Insuficiente	No Satisfactorio	1	☹
Parcial	Poco Satisfactorio	2	☹☹
Suficiente	Satisfactorio	3	☺☺☺
Excelente	Muy Satisfactorio	4	☺☺☺☺

Para la realización de la comparación se utilizara la siguiente nomenclatura:

*U* = Representa el puntaje obtenido por el LMS Moodle.

*V* = Representa el puntaje obtenido por el LMS Dokeos.

*X* = Representa el puntaje obtenido por el LMS ATutor.

*Y* = Representa el puntaje obtenido por el LMS Claroline.

*Z* = Representa el puntaje obtenido por el LMS Sakai.

*W* = Representa el puntaje sobre el cual será evaluado el parámetro.

*Cm* = Representa el puntaje alcanzado de Moodle en el parámetro.

***Cd** = Representa el puntaje alcanzado de Dokeos en el parámetro.*

***Ca** = Representa el puntaje alcanzado de ATutor en el parámetro.*

***Cc** = Representa el puntaje alcanzado de Claroline en el parámetro.*

***Cs** = Representa el puntaje alcanzado de Sakai en el parámetro.*

***Ct** = Representa el puntaje por el cual es evaluado el parámetro.*

***Pm** = Calificación porcentual obtenida por Moodle.*

***Pd** = Calificación porcentual obtenida por Dokeos.*

***Pa** = Calificación porcentual obtenida por ATutor.*

***Pc** = Calificación porcentual obtenida por Claroline.*

***Ps** = Calificación porcentual obtenida por Sakai.*

Las fórmulas que se utilizaran en el proceso del análisis comparativo son las siguientes:

$$Cm = \sum U$$

$$Cd = \sum V$$

$$Ca = \sum X$$

$$Cc = \sum Y$$

$$Cs = \sum Z$$

$$Ct = \sum W$$

$$Pm = \left(\frac{Cm}{Ct}\right) * 100\%$$

$$Pd = \left(\frac{Cd}{Ct}\right) * 100\%$$

$$Pa = \left(\frac{Ca}{Ct}\right) * 100\%$$

$$Pc = \left(\frac{Cc}{Ct}\right) * 100\%$$

$$Ps = \left(\frac{Cs}{Ct}\right) * 100\%$$

### 2.4.3.1. Establecimiento de Parámetros

#### 2.4.3.1.1. Madurez del producto

- **Indicador 1.1: Idiomas:** Este indicador se medirá de acuerdo al número de idiomas que soportan los LMS.

Tabla II. VII: Valoración del Indicador 1.1: Idiomas

Numero de Idiomas	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
10 a 35	1	No Satisfactorio
36 a 55	2	Poco Satisfactorio
56 a 75	3	Satisfactorio
76 a 95	4	Muy Satisfactorio

- **Indicador 1.2: Plataforma:** En este indicador se valorizará de acuerdo al número de plataformas que soportan los LMS.

Tabla II. VIII: Valoración del Indicador 1.2: Plataformas

Numero de Plataformas	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
1	1	No Satisfactorio
2	2	Poco Satisfactorio
3	3	Satisfactorio
4	4	Muy Satisfactorio

- **Indicador 1.3: Base de datos:** En este indicador se valorizará de acuerdo al número de Base de Datos que soportan los LMS.

Tabla II. IX: Valoración del Indicador 1.3: Base de Datos

Número de Base de Datos	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
1	1	No Satisfactorio
2	2	Poco Satisfactorio
3	3	Satisfactorio
4	4	Muy Satisfactorio

- **Indicador 1.4: Servidor Web:** En este indicador se valorizará de acuerdo al Servidor Web que soportan los LMS.

Tabla II. X: Valoración del Indicador 1.4: Servidor web

Numero de Servidores Web	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
1	4	No Satisfactorio
2	3	Poco Satisfactorio
3	2	Satisfactorio
4	1	Muy Satisfactorio

- **Indicador 1.5: Lenguaje de Programación:** En este indicador se valorizará de acuerdo los lenguajes de programación en los que son desarrollados los LMS.

Tabla II. XI: Valoración del Indicador 1.5: Lenguaje de Programación

Número de lenguajes de programación	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
1	1	No Satisfactorio
2	2	Poco Satisfactorio
3	3	Satisfactorio

Valoración		
4	4	Muy Satisfactorio

#### 2.4.3.1.1.1. Valoraciones

Tabla II. XII: Resultados del Parámetro 1: Madurez del Producto

	LMS	Idiomas	Plataforma	Base de datos	Servidor Web	Lenguaje de programación
MOODLE	Valor cualitativo	Muy satisfactorio	Muy Satisfactorio	Muy Satisfactorio	Muy Satisfactorio	No Satisfactorio
	Valor Cuantitativo	4	4	4	4	1
DOKEOS	Valor cualitativo	Poco Satisfactorio	Muy Satisfactorio	No Satisfactorio	Muy Satisfactorio	No Satisfactorio
	Valor Cuantitativo	2	4	1	4	1
ATUTOR	Valor cualitativo	No Satisfactorio	Muy Satisfactorio	No Satisfactorio	Muy Satisfactorio	Poco Satisfactorio
	Valor Cuantitativo	1	4	1	4	2
CLAROLINE	Valor cualitativo	No Satisfactorio	Muy Satisfactorio	No Satisfactorio	Muy Satisfactorio	No Satisfactorio
	Valor Cuantitativo	1	4	1	4	1
SAKAI	Valor cualitativo	No Satisfactorio	Muy Satisfactorio	Poco Satisfactorio	Muy Satisfactorio	No Satisfactorio
	Valor Cuantitativo	1	4	2	4	1

#### 2.4.3.1.1.2. Interpretación

**Idioma:** Este indicador es muy importante ya que nos representa los lenguajes en los que puede ser utilizadas las diferentes herramientas LMS. Moodle presenta una gran ventaja ante los otros LMS ya que soporta 86 idiomas, obteniendo 4 puntos correspondiente a Muy satisfactorio. Dokeos con 40 idiomas, obteniendo 2 puntos equivalentes a poco Satisfactorio. Claroline con 35 idiomas, ATutor soporta 32 idiomas y Sakai cuenta con un soporte bajo de idiomas por lo que obtienen 1 punto equivalentes a No Satisfactorio.

**Plataforma:** Al analizar este indicador se obtuvo que las herramientas LMS analizadas son multiplataforma soportando plataformas LAMP (Linux, Apache, My SQL, PHP), WAMP (Windows, Apache, My SQL, PHP), Unix, Mac OS x. Obteniendo por lo tanto todos las Herramientas LMS tienen 4 puntos equivalentes a Muy Satisfactorio.

**Motor de Base de Datos:** Es un indicador de suma importancia ya que aquí se indica con cuantos motores de base de Datos pueden soportar los LMS. Moodle es una herramienta que soporta multiples Motores de Base de Datos como PostgreSQL, MySQL, Oracle, MSSQL obteniendo una calificación de 4 equivalente a muy Satisfactorio. Sakai soporta MySQL y Oracle como motor de Base de Datos obteniendo una calificación de 2 equivalente a Poco Satisfactorio. Dokeos, ATutor y Claroline soportan como motor de Base de Datos únicamente a MySQL por lo que su calificación es 1 equivalente a No Satisfactorio.

**Servidor Web:** Este indicador indica en que servidor Web sera alodo la Herramienta LMS los 5 LMS soportan Apache como servidor Web por lo que obtienen una calificación de 4 puntos equivalentes a Muy Satisfactorio.

**Lenguaje de programación:** Moodle, Dokeos y Claroline son desarrollados en el lenguaje PHP por lo que obtiene una calificación de 1 equivalente a No Satisfactorio. Sakai es desarrollado en Java por lo que tambien obtiene una calificación de 1. Y ATutor es desarrollado tanto en php como en java por lo que obtiene una calificación de 2 equivalente a Poco Satisfactorio.

#### 2.4.3.1.1.3. Calificación

**Calculo de los porcentajes.**

$$Cm = \sum X$$

$$Cd = \sum V$$

$$Ca = \sum X$$

$$Cc = \sum Y$$

$$Cs = \sum Z$$

$$Ct = \sum W$$

$$Pm = \left(\frac{Cm}{Ct}\right) * 100\%$$

$$Pd = \left(\frac{Cd}{Ct}\right) * 100\%$$

$$Pa = \left(\frac{Ca}{Ct}\right) * 100\%$$

$$Pc = \left(\frac{Cc}{Ct}\right) * 100\%$$

$$Ps = \left(\frac{Cs}{Ct}\right) * 100\%$$

$$Cm: 4 + 4 + 4 + 4 + 1 = 17$$

$$Cd: 2 + 4 + 1 + 4 + 1 = 12$$

$$Ca: 1 + 4 + 1 + 4 + 2 = 12$$

$$Cc: 1 + 4 + 1 + 4 + 1 = 11$$

$$Cs: 1 + 4 + 2 + 4 + 1 = 11$$

$$Ct: 4 + 4 + 4 + 4 + 4 = 20$$

$$Pm: \left(\frac{17}{20}\right) * 100\% = 85\%$$

$$Pd: \left(\frac{12}{20}\right) * 100\% = 60\%$$

$$Pa: \left(\frac{12}{20}\right) * 100\% = 60\%$$

$$Pc: \left(\frac{11}{20}\right) * 100\% = 55\%$$

$$Ps: \left(\frac{11}{20}\right) * 100\% = 55\%$$



Ilustración II. 8: Resultado final Parámetro 1- Madurez del Producto

#### 2.4.3.1.1.4. Representación de Resultados

Tabla II. XIII: Representación de resultados

LMS		INDICADORES				
		Idiomas	Plataforma	Base de datos	Servidor Web	Lenguaje de programación
MOODLE	Valor Calificativo	Excelente	Excelente	Excelente	Excelente	Insuficiente
	Valor Representativo	☺☺☺☺	☺☺☺☺	☺☺☺☺	☺☺☺☺	☺
DOKEOS	Valor Calificativo	Parcial	Excelente	Insuficiente	Excelente	No Satisfactorio
	Valor Representativo	☺☺	☺☺☺☺	☺	☺☺☺☺	☺
ATUTOR	Valor Calificativo	Insuficiente	Excelente	Insuficiente	Excelente	Parcial
	Valor Representativo	☺	☺☺☺☺	☺	☺☺☺☺	☺☺
CLAROLINE	Valor Calificativo	Insuficiente	Excelente	Insuficiente	Excelente	Insuficiente
	Valor Representativo	☺	☺☺☺☺	☺	☺☺☺☺	☺
SAKAI	Valor Calificativo	Insuficiente	Excelente	Parcial	Excelente	Insuficiente
	Valor Representativo	☺	☺☺☺☺	☺☺	☺☺☺☺	☺

### 2.4.3.1.2. Personalización

- **Indicador 2.1: Personalización de Apariencia:** Este indicador se basará en si la plataforma permite o no la personalización de sus temas y por ende la agregación de nuevos temas.

Tabla II. XIV: Valoración del Parámetro 2 - Indicador 2.1 Apariencia

Creación/Personalización	Valoración	
	Valoración Cuantitativa /4	Valoración cualitativa
Insuficiente	1	No Satisfactorio
Parcial	2	Poco Satisfactorio
Suficiente	3	Satisfactorio
Excelente	4	Muy Satisfactorio

- **Indicador 2.2: Niveles de acceso:** Es éste indicador se basará en el número de niveles de acceso o roles que poseen las herramientas LMS.

Tabla II. XV: Valoración del Parámetro 2 - Indicador 2.2 Niveles de Acceso

Número de niveles de Acceso	Valoración	
	Valoración Cuantitativa /4	Valoración cualitativa
1 a 2	1	No Satisfactorio
3 a 4	2	Poco Satisfactorio
5 a 6	3	Satisfactorio
6 a 7	4	Muy Satisfactorio

- **Indicador 2.3: Documentación:** Este indicador se calificará en base a la documentación existe para ayuda y guía de los desarrolladores.

Tabla II. XVI: Valoración del Parámetro 2 - Indicador 2.3 Documentación

Cantidad de Documentación	Valoración	
	Valoración Cuantitativa /4	Valoración cualitativa
Nula	1	No Satisfactorio
Escasa	2	Poco Satisfactorio
Poca	3	Satisfactorio
Suficiente	4	Muy Satisfactorio

- **Indicador 2.4: Estándares:** Éste indicador se medirá de acuerdo al cumplimiento de estándares por parte de los LMS.

Tabla II. XVII: Valoración del Parámetro 2 - Indicador 2.4 Estándares

Cumplimiento	Valoración	
	Valoración Cuantitativa /4	Valoración cualitativa
Insuficiente	1	No Satisfactorio
Parcial	2	Poco Satisfactorio
Suficiente	3	Satisfactorio
Excelente	4	Muy Satisfactorio

- **Indicador 2.5: Personalización y Creación de nuevos módulos:** Este índice se cuantificará de acuerdo a la Capacidad de personalizar y creación de nuevos componentes de los LMS.

Tabla II. XVIII: Valoración del Parámetro 2 - Indicador 2.5 Personalización y Creación

Creación/Personalización	Valoración	
	Valoración Cuantitativa /4	Valoración cualitativa
Insuficiente	1	No Satisfactorio

Valoración		
Parcial	2	Poco Satisfactorio
Suficiente	3	Satisfactorio
Excelente	4	Muy Satisfactorio

#### 2.4.3.1.2.1. Valoración

Tabla II. XIX. Valoración de los Indicadores del Parámetro 2– Personalización

LMS		INDICADORES				
		Personalización de Apariencia	Niveles de Acceso	Documentación	Estándares	Personalización y Creación
MOODLE	Valor cualitativo	Muy satisfactorio	Muy satisfactorio	Satisfactorio	Muy Satisfactorio	Muy Satisfactorio
	Valor Cuantitativo	4	4	3	4	4
DOKEOS	Valor cualitativo	No Satisfactorio	Poco Satisfactorio	Satisfactorio	Satisfactorio	Satisfactorio
	Valor Cuantitativo	1	2	3	3	3
ATUTOR	Valor cualitativo	No Satisfactorio	No Satisfactorio	Poco Satisfactorio	Muy Satisfactorio	No Satisfactorio
	Valor Cuantitativo	1	1	2	4	1
CLAROLINE	Valor cualitativo	No Satisfactorio	No Satisfactorio	Poco Satisfactorio	Satisfactorio	Poco Satisfactorio
	Valor Cuantitativo	1	1	2	3	2
SAKAI	Valor cualitativo	No Satisfactorio	No Satisfactorio	No Satisfactorio	No Satisfactorio	No Satisfactorio
	Valor Cuantitativo	1	1	1	1	1

#### 2.4.3.1.2.2. Interpretación

Los indicadores descritos en el parámetro Personalización son claves para la determinación de las dos principales herramientas LMS que permitan realizar agregación de componentes para su posterior análisis.

**Personalización de apariencia:** En éste indicador se tiene que Moodle obtuvo 4 puntos ya que permite la creación de nuevos temas y por ende permite la personalización de su apariencia. Dokeos, ATutor, Claroline y Sakai no permiten la personalización ni la creación de nuevos temas por lo que obtuvieron 1 punto equivalente a no Satisfactorio.

**Niveles de Acceso:** Los niveles de acceso permiten determinar si un usuario está o no en capacidad de agregar un nuevo componente o personalizar un ya existe. Moodle es una herramienta que posee 7 niveles de acceso (Administrador, Docente, Estudiante, Invitado, Profesor no editor, Creador de cursos, Usuario autenticado) además permite la creación de nuevos roles y establece capacidades de acuerdo a las necesidades de los usuarios es por ello que su calificación es 4 equivalente a Muy Satisfactorio. Dokeos por su lado tiene 4 niveles de usuario (Administrador, Estudiante, Profesor y creador de cursos) pero no permite la creación de nuevos roles o niveles de acceso por ello tiene 2 puntos equivalente a poco satisfactorio, ATutor tienen únicamente 3 roles (Administrador, Estudiante y docente) y no permite la creación de nuevos roles, Claroline y Sakai poseen 2 roles Estudiante y Docente y tampoco permiten la agregación de nuevos roles por lo que tienen 1 punto equivalente a No Satisfactorio.

**Estándares:** En este indicador Moodle y ATutor poseen 4 puntos equivalentes a Muy Satisfactorio ya que están adaptados a los estándares más importantes como AICC y el SCORM para la agregación de los contenidos de formación a la plataforma, ATutor además cuenta con estándares de accesibilidad de W3C WCAG 1.0 en la AA+. Dokeos y Claroline respeta los estándares SCORM obteniendo 2 puntos equivalentes a Poco Satisfactorio. Sakai presenta escaso soporte a estándares por lo que obtiene un punto equivalente a no Satisfactorio.

**Personalización y Creación de nuevos componentes:** Este indicador es fundamental a la hora de elegir los LMS más adecuados a la agregación de nuevos componentes. Moodle posee 4 puntos ya que esta herramienta permite la personalización de componentes existentes y la creación y por ende la agregación de los mismos de acuerdo a las necesidades de los programadores ayudando a mejora la plataforma. Como programadores se tiene la capacidad de controlar el acceso de las nuevas

funcionalidades. Dokeos permite la agregación de nuevos componentes pero no la personalización de los componentes existentes por esta razón obtiene 3 puntos equivalentes a Satisfactorio. ATutor, Claroline y Sakai son plataformas que no permiten la agregación y colaboración por parte de desarrolladores siendo plataformas nuevas con escasas en documentación para desarrolladores orientadas a la enseñanza y colaboración del sistema educativo obteniendo 1 punto equivalentes a No Satisfactorio.

#### 2.4.3.1.2.3. Calificación

##### Cálculo de los porcentajes.

$$Cm = \sum U$$

$$Cd = \sum V$$

$$Ca = \sum X$$

$$Cc = \sum Y$$

$$Cs = \sum Z$$

$$Ct = \sum W$$

$$Pm = \left(\frac{Cm}{Ct}\right) * 100\%$$

$$Pd = \left(\frac{Cd}{Ct}\right) * 100\%$$

$$Pa = \left(\frac{Ca}{Ct}\right) * 100\%$$

$$Pc = \left(\frac{Cc}{Ct}\right) * 100\%$$

$$Ps = \left(\frac{Cs}{Ct}\right) * 100\%$$

$$Cm: 4 + 4 + 3 + 4 + 4 = 19$$

$$Cd: 1 + 2 + 3 + 3 + 3 = 12$$

$$Ca: 1 + 1 + 2 + 4 + 1 = 9$$

$$Cc: 1 + 1 + 2 + 3 + 2 = 9$$

$$Cs: 1 + 1 + 1 + 1 + 1 = 5$$

$$Ct: 4 + 4 + 4 + 4 + 4 = 20$$

$$Pm: \left(\frac{19}{20}\right) * 100\% = 95\%$$

$$Pd: \left(\frac{12}{20}\right) * 100\% = 60\%$$

$$Pa: \left(\frac{9}{20}\right) * 100\% = 45\%$$

$$Pc: \left(\frac{9}{20}\right) * 100\% = 45\%$$

$$Ps: \left(\frac{5}{20}\right) * 100\% = 25\%$$



Ilustración II. 9: Resultado final Parámetro 1- Personalización y Creación

#### 2.4.3.1.2.4. Representación de Resultados

Tabla II. XX: Representación de Resultados Parámetro 2

LMS		INDICADORES				
		Personalización De Apariencia	Niveles de Acceso	Documentación	Estándares	Personalización y Creación
MOODLE	Valor Calificativo	Excelente	Excelente	Suficiente	Excelente	Excelente
	Valor Representativo	😊😊😊😊	😊😊😊😊	😊😊😊	😊😊😊😊	😊😊😊😊
DOKEOS	Valor Calificativo	Insuficiente	Parcial	Suficiente	Suficiente	Suficiente
	Valor Representativo	😊	😊😊	😊😊😊	😊😊😊	😊😊😊

		INDICADORES				
ATUTOR	Valor Calificativo	Insuficiente	Insuficiente	Parcial	Excelente	Insuficiente
	Valor Representativo	☺	☺	☺☺	☺☺☺☺	☺
CLAROLINE	Valor Calificativo	Insuficiente	Insuficiente	Parcial	Suficiente	Parcial
	Valor Representativo	☺	☺	☺☺	☺☺☺	☺☺
SAKAI	Valor Calificativo	Insuficiente	Insuficiente	Insuficiente	Insuficiente	Insuficiente
	Valor Representativo	☺	☺	☺	☺	☺

### 2.4.3.2. Puntajes alcanzados

Tabla II. XXI: Cuadro de Resultados Parámetro - Indicador

		Herramientas LMS				
Parámetros e Indicadores		Moodle	Doceos	ATutor	Claroline	Sakai
Madurez del Producto	Idiomas	4	2	1	1	1
	Plataformas	4	4	4	4	4
	Motores de Base de datos	4	1	1	1	2
	Servidor Web	4	4	4	4	4
	Lenguajes de programación	1	1	2	1	1
Personalización	Personalización de la Apariencia	4	1	1	1	1
	Niveles de Acceso	4	2	1	1	1
	Documentación	3	3	2	2	1
	Estándares	4	3	4	2	1
	Personalización y Creación	4	3	1	2	1
<b>Total</b>		<b>36</b>	<b>24</b>	<b>21</b>	<b>19</b>	<b>17</b>

Tabla II. XXII: Resultados Generales

Herramientas LMS					
Parámetros	Moodle	Dokeos	ATutor	Claroline	Sakai
Madurez del Producto	85%	60%	60%	55%	55%
Personalización	95%	60%	45%	45%	25%
Total %	90%	60%	52,5%	50%	40%

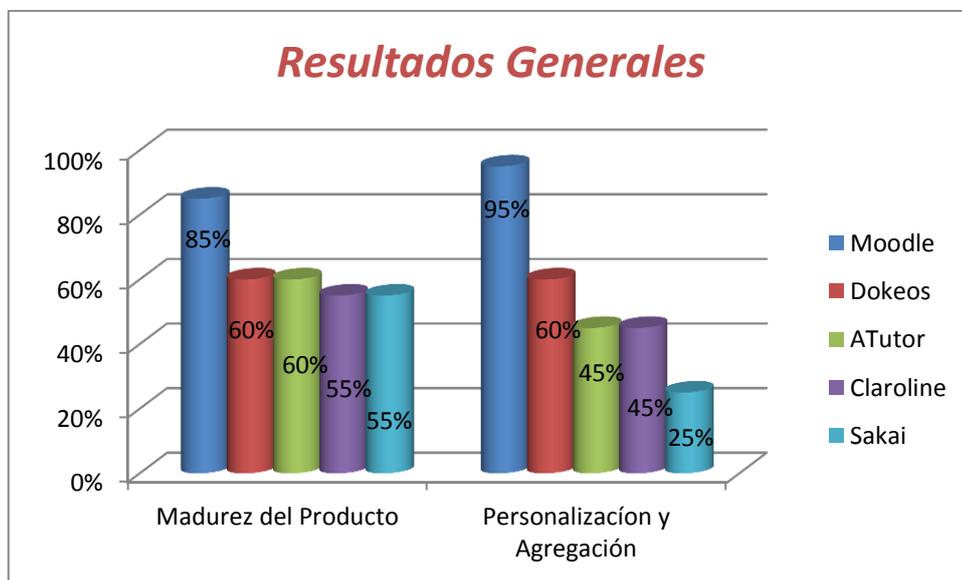


Ilustración II. 10: Representación Gráfica de Resultados Generales

### 2.4.3.3. Interpretación:

Con la tabulación de los resultados obtenidos en la evaluación de los LMS, se da a conocer de una manera global el comportamiento de cada LMS respecto a la agregación de componentes.

El parámetro Madurez del Producto fue el encargado de probar aspectos o características básicas de los LMS. Destacándose de entre las 5 herramientas LMS Moodle con un 85%, Dokeos y ATutor con el mismo porcentaje de 60%.

El parámetro Personalización y Creación permitió determinar si los LMS permiten la personalización de componentes existentes y la creación y por ende la agregación de los mismos, destacándose de entre los cinco LMS Moodle y Dokeos con el 95% y el 60% respectivamente mostrando de esta manera que las dos herramientas son las más aptas en la agregación de componentes.

#### 2.4.3.4. Análisis de Resultados

Los resultados que presenta cada LMS al establecer mismos criterios son claros ya que las cinco herramientas presentan características similares se puede establecer los LMS que se adaptan de mejor manera a la agregación de componentes.

Luego de la interpretación de los resultados realizado anteriormente se destaca los LMS Moodle y Dokeos para la agregación de componentes, ideales para desarrollar el análisis comparativo de cuan Productivo resulta la agregación de componentes en las dos herramientas basadas en parámetros y prototipos de componentes agregados en las herramientas mencionadas.

Tabla II. XXIII: Valores y Porcentajes Finales

	TOTAL	
	Valor	Porcentaje
<b>Moodle</b>	36	90
<b>Dokeos</b>	24	60
<b>ATutor</b>	21	52,5
<b>Claroline</b>	19	50
<b>Sakai</b>	17	40



Ilustración II. 11: Resultado final del Análisis

Haciendo referencia a la Tabla XXIII y por observación directa se concluye que las plataformas LMS más óptimas para el análisis de la productividad en la agregación de componentes son: Moodle posee el valor más alto 40 y Dokeos 28 puntos sobre los 25, 23 y 21 de ATutor, Claroline y Sakai respectivamente. Razón por la cual se analizará la productividad en la agregación de componentes en Moodle y Dokeos.

## 2.5. EL MODELO SCRUM

### 2.5.1. Origen

SCRUM es una metodología ágil de desarrollo de proyectos que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujijo Nonaka a mediados de los 80.

Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software. En el desarrollo de software SCRUM está considerado como modelo ágil por la Agile Alliance.

### 2.5.2. Introducción al modelo

SCRUM es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. SCRUM es una metodología ágil, y como tal:

Es un modo de desarrollo de carácter adaptable más que predictivo.

- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

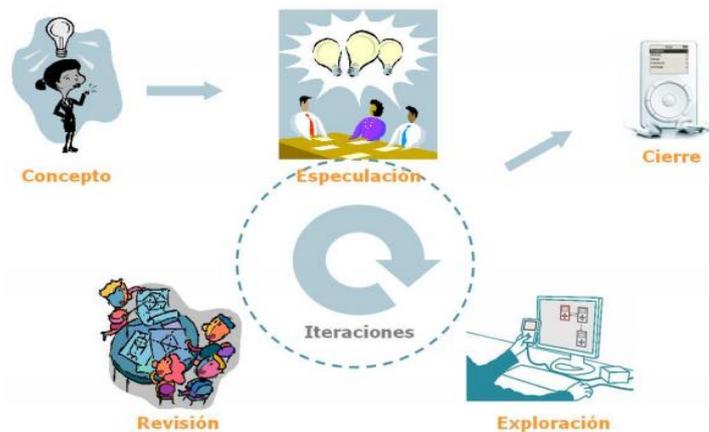


Ilustración II. 12: Estructura del desarrollo ágil.

Se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve (normalmente de 30 días).

Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto.

Estas iteraciones son la base del desarrollo ágil, y SCRUM gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente.

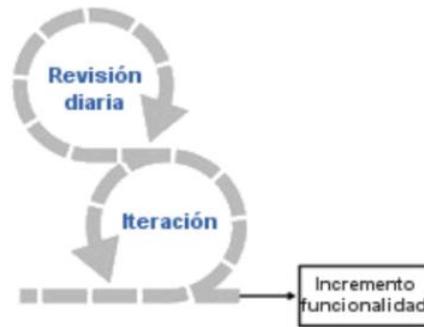


Ilustración II. 13: Estructura central de SCRUM

### 2.5.3. Control de la evolución del proyecto

SCRUM controla de forma empírica y adaptable la evolución del proyecto, empleando las siguientes prácticas de la gestión ágil:

### 2.5.4. Revisión de las Iteraciones

Al finalizar cada iteración (normalmente 30 días) se lleva a cabo una revisión con todas las personas implicadas en el proyecto. Este es el periodo máximo que se tarda en reconducir una desviación en el proyecto o en las circunstancias del producto

### 2.5.5. Desarrollo incremental

Durante el proyecto, las personas implicadas no trabajan con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.

### 2.5.6. Desarrollo evolutivo

Los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad de requisitos.

Intentar predecir en las fases iniciales cómo será el producto final, y sobre dicha predicción desarrollar el diseño y la arquitectura del producto no es realista, porque las circunstancias obligarán a remodelarlo muchas veces.

Para qué predecir los estados finales de la arquitectura o del diseño si van a estar cambiando. En SCRUM se toma a la inestabilidad como una premisa, y se adoptan técnicas de trabajo para permitir esa evolución sin degradar la calidad de la arquitectura que se irá generando durante el desarrollo.

El desarrollo SCRUM va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto. No los considera como productos que deban realizarse en la primera “fase” del proyecto.

(El desarrollo ágil no es un desarrollo en fases)

#### **2.5.7. Auto-organización**

Durante el desarrollo de un proyecto son muchos los factores impredecibles que surgen en todas las áreas y niveles. La gestión predictiva confía la responsabilidad de su resolución al gestor de proyectos.

En SCRUM los equipos son auto-organizados (no auto-dirigidos), con margen de decisión suficiente para tomar las decisiones que consideren oportunas.

#### **2.5.8. Colaboración**

Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo. Ésta es necesaria, porque para que funcione la auto-organización como un control eficaz cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

#### **2.5.9. Visión general del proceso**

SCRUM denomina “sprint” a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días. El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental.

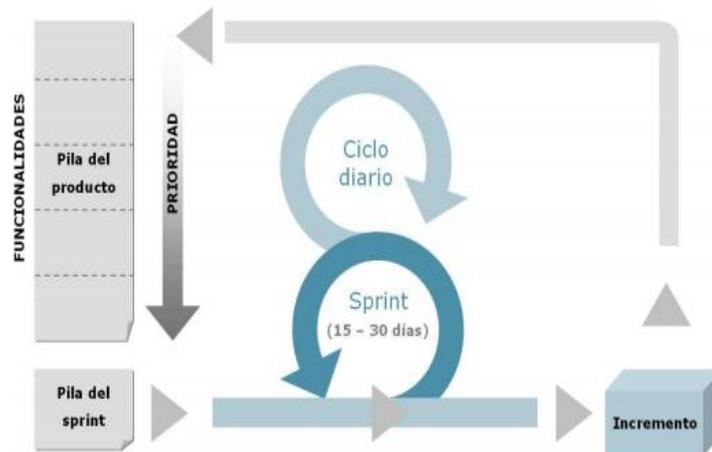


Ilustración II. 14: Sprint del SCRUM

Los elementos que conforman el desarrollo SCRUM son:

➤ **Las reuniones**

- **Planificación de sprint:** Jornada de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.
- **Reunión diaria:** Breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.
- **Revisión de sprint:** Análisis y revisión del incremento generado.

➤ **Los elementos**

- **Pila del producto:** lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo y evolucionando durante el desarrollo.
- **Pila del sprint:** Lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- **Incremento:** Resultado de cada sprint

## **CAPITULO III**

### **ANÁLISIS DE LA PRODUCTIVIDAD EN LA AGREGACIÓN DE MÓDULOS**

En este capítulo se determinará los parámetros en base a la agregación de módulos o componentes prototipos en ambas herramientas. Para lo cual es necesario estudiar con anterioridad a profundidad las herramientas seleccionadas que son Moodle y Dokeos su instalación, estructura, características, Base de datos, Arquitectura, etc... También se expondrá el Lenguaje PHP, qué es y sus características, ya que es el lenguaje en el cuál se encuentran desarrolladas Moodle como Dokeos.

Luego de haber establecido los parámetros realizaremos la comparación, el respectivo análisis entre ambas herramientas y determinaremos cuál es la herramienta más productiva en la agregación de Componentes.

### **3.1. MOODLE [1]**

Moodle se encuadra dentro de la categoría de los LMS, y puede ser utilizado tanto para desarrollar cursos completamente on-line siguiendo la metodología del e-learning, como para complementar la educación presencial haciendo uso tan solo de las herramientas que el profesor considere oportunas para facilitar sus clases (metodología b-learning).

La palabra Moodle es el acrónimo de Modular Object-Oriented Dynamic Learning Environment (Ambiente de Aprendizaje Modular Orientado a Objetos Dinámicos). Se distribuye gratuitamente bajo licencia GPL de GNU.

Entre las características más importantes de la plataforma Moodle a nivel técnico destacan las siguientes:

1. Su diseño es modular, permitiendo gran flexibilidad para agregar y suprimir funcionalidades en muchos niveles.
2. Se ejecuta sin necesidad de cambios en el sistema operativo bajo Unix, Linux, Windows, Mac OS X, Netware y todos aquellos sistemas operativos que permitan PHP.
3. Soporta las principales marcas de manejadores de bases de datos.
4. Su actualización desde una versión anterior a la siguiente es un proceso muy sencillo. Dispone de un sistema interno capaz de reparar y actualizar sus bases de datos cada cierto tiempo.

A nivel pedagógico Moodle ofrece funcionalidades bastante atractivas para los docentes, como las siguientes:

1. Moodle se rige por una filosofía basada en la teoría constructivista, sustentada en la idea de interacción con los demás y a través de la conexión de conocimientos nuevos con aprendizaje previamente adquirido. Conectando la estrategia didáctica del profesorado con las ideas previas del alumnado, construyendo de manera sólida

los conceptos, interconectando los unos con los otros en forma de red de conocimiento.

2. Cuenta con una interfaz atractiva, de tecnología sencilla, ligera eficiente y compatible.
3. Permite el acceso de invitados a los cursos.
4. Los cursos son clasificados en categorías, lo que facilita su búsqueda.
5. Es fácil de instalar, por lo que no precisa un nivel avanzado de conocimientos informáticos para proceder a su implementación.
6. Un sitio Moodle puede albergar miles de cursos y estos pueden ser clasificados en distintas categorías.

Moodle posee una interfaz intuitiva que facilita a los diseñadores de cursos la construcción de foros, chats, lista de correo, glosarios, bibliotecas, autoevaluaciones y demás funcionalidades. Los recursos disponibles en Moodle se pueden clasificar en tres categorías: Recursos Transmisivos, Recursos Interactivos y Recursos Colaborativos. A estas tres categorías también podemos añadir las herramientas de comunicación.

- **Recursos Transmisivos:** Se refiere a todos los componentes (módulos, recursos, actividades) en Moodle que tienen como función principal la de transmitir información. El profesor se establece prácticamente en emisor único. Los alumnos, en receptores. Lo más usual es que estos contenidos estén constituidos por algún tipo de texto más o menos estructurado con apoyo de imágenes, esquemas. Los recursos Transmisivos disponibles en Moodle son: Página de texto, Página Web, Enlace a archivo o Web, Directorio, Etiqueta y Libro.
- **Recursos interactivos:** Estos recursos se centran más en el alumno, quien tiene determinado control de navegación sobre los contenidos. Cuanto menos lineales sean los contenidos y la propia navegación, mayor interactividad habrá. En Moodle los Recursos interactivos disponibles son: Lecciones, Cuestionarios, SCORM, Glosarios y Tareas.

- **Recursos colaborativos:** Estas herramientas pueden introducir en nuestras escuelas la posibilidad de disponer de recursos altamente orientados a la interacción y el intercambio de ideas y materiales tanto entre el profesor y los alumnos como de los alumnos entre sí Moodle incluye las siguientes herramientas colaborativas: Foros, Talleres y Wikis.
  
- **Herramientas de comunicación:** Uno de los propósitos principales de la plataforma Moodle consiste en facilitar y enriquecer la interacción entre todos los miembros de la comunidad (estudiantes entre sí, profesores entre sí y entre alumnos y profesores). Por ello las herramientas de comunicación son básicas. Las herramientas de comunicación son un requisito para que las otras tres funcionen adecuadamente. Las herramientas de comunicación disponibles en Moodle son: Correo electrónico, Chats, Mensajes, Consultas y Encuestas.

Una de las características principales de Moodle es su modularidad. Cada uno de los recursos que componen la plataforma se define como una pieza de software independiente denominados COMPONENTES o MODULOS que puede ser modificado o eliminado según las necesidades de la institución de enseñanza que la adopte. De la misma forma también es posible instalar nuevos MODULOS o COMPONENTES que añadan funcionalidad al sistema, bien descargándolos desde la página oficial de módulos no estándar de Moodle o bien desarrollándolos a medida. El proceso de desarrollo de un módulo no es complejo, pero requiere conocer un poco más a fondo la arquitectura y el funcionamiento interno del sistema. Por ello, antes de lanzarnos al desarrollo de un nuevo módulo, estudiaremos con detenimiento dicha estructura interna de Moodle.

### **3.1.1. Arquitectura de Moodle [3]**

Moodle significa Modular Object-Oriented Dynamic Learning Environment (Entorno de enseñanza dinámico orientado a objetos modular). Desde la perspectiva del programador, la “M” de Moodle contiene un concepto muy importante, ya que Moodle, en su totalidad, está diseñado de forma modular. Esto permite al desarrollador realizar importantes modificaciones a la plataforma sin necesidad de cambiar el código fuente.

La modularidad se convierte en una importante característica que permite reducir la cantidad de tiempo empleada en realizar modificaciones cuando se lanzan nuevas versiones de Moodle.

### 3.1.1.1. Entorno de Moodle

Moodle es un ejemplo de aplicación LAMP (Linux, Apache, MySQL y PHP). Se trata fundamentalmente de aplicaciones web escritas con un lenguaje de script, que utilizan una base de datos SQL para almacenar información. Con el incremento de la popularidad de ejecutar aplicaciones web de código abierto en Windows y en Mac OS X, han surgido dos nuevos términos: WAMP y MAMP. Para el desarrollo de este proyecto de tesis utilizaremos WAMP (Windows, Apache, MySQL y PHP). En la siguiente figura se muestran los componentes del sistema en Moodle.

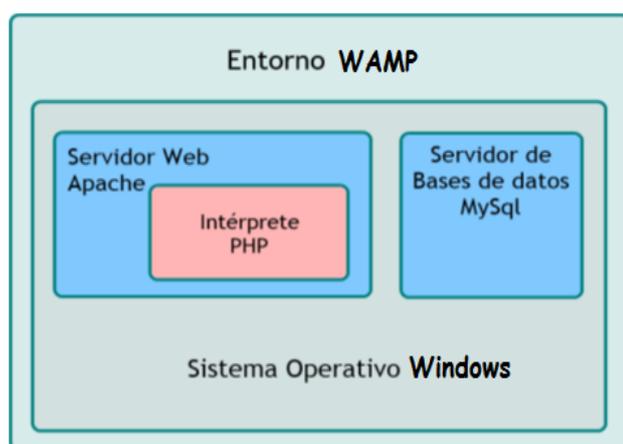


Ilustración III. 1: Entorno Moodle (WAMP)

Moodle está escrito en PHP, y la versión 1.9 requiere la versión de PHP 4.3.0 o superior para funcionar. La versión 2.0 o más de Moodle, necesitará la versión de PHP 5.2.8 o superior.

### 3.1.1.2. Base de datos

La capa de base de datos en Moodle ha sido escrita utilizando la librería de PHP ADOdb, que fue creada para proporcionar un método estándar de acceso a varios

sistemas de bases de datos, utilizando una interfaz de programación consistente. Gracias al uso de ADOdb, Moodle proporciona soporte para varias bases de datos, entre las que se incluyen: MySQL, PostgreSQL, Microsoft SQL y Oracle. Aunque Moodle proporciona soporte para muchas bases de datos, en la práctica la mayoría de los sistemas están desplegados utilizando MySQL, por lo que se presta más atención a las instalaciones de MySQL, que por este motivo tienen menos fallos y cuentan con el conjunto más amplio de extensiones de terceros.

### **3.1.1.3. Sistema operativo**

Moodle es capaz de funcionar en cualquier sistema operativo que soporte la versión requerida de PHP y la base de datos. Generalmente se instala en alguno de los tres principales sistemas operativos: Windows, Mac OS X o Linux (o sistemas operativos basados en Unix). Si los comparamos con Linux, Mac OS X y Windows tienen un menor rendimiento en desarrollos a gran escala.

### **3.1.1.4. Servidor Web**

Moodle normalmente trabajará con cualquier servidor web que soporte la versión apropiada de PHP. En la práctica, el servidor web más usado es Apache, que está disponible para la mayoría de sistemas operativos. Internet Information Services (IIS) es otro servidor web muy popular para alojar Moodle.

Moodle, como aplicación web, tiene sus propias limitaciones de soporte y desarrollo, debido al modo en que se usa. Esto significa que, como desarrolladores, tenemos que conocer el tipo de arquitecturas con las que serán usadas nuestras modificaciones. También significa que tenemos que ser conscientes de las implicaciones de rendimiento del código que se añade.

### 3.1.1.5. Estructura de directorios

La siguiente pantalla es el listado de los directorios de una instalación reciente de Moodle 2.3.0.

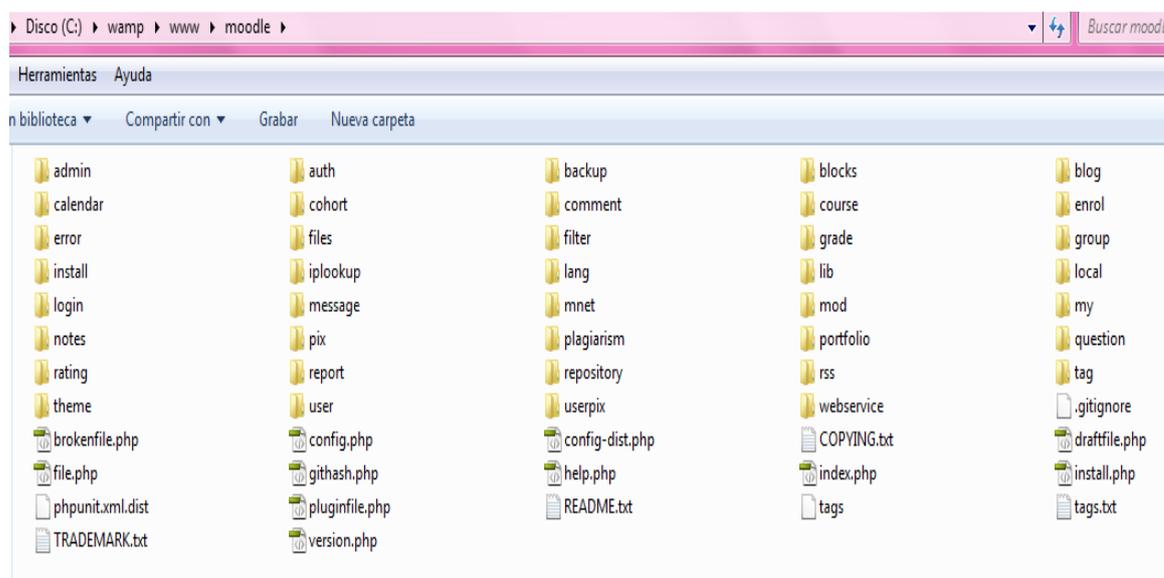


Ilustración III. 2: Directorios de Moodle

A continuación, exploraremos algunos de los directorios más importantes utilizados por los desarrolladores que desean realizar cambios a Moodle. Moodle utiliza una nomenclatura simple para sus módulos, cuyos archivos se almacenan en su propia carpeta, y el nombre de esta carpeta es el nombre que Moodle muestra en su interfaz cuando se referencia al módulo.

- **admin:** Esta carpeta almacena los ficheros PHP que controlan la interfaz de los usuarios administradores
- **auth:** La carpeta *auth* contiene todos los módulos de autenticación de Moodle. Cada módulo tendrá su propio directorio dentro de esta carpeta. Los módulos de autenticación controlan la creación de usuarios y el acceso de estos al sistema.
- **backup:** Esta carpeta contiene las utilidades de copia de seguridad del núcleo del sistema. Se trata de funciones para guardar, restaurar e importar cursos.
- **blocks:** Los bloques se usan para mostrar bloques de información en la columna de la izquierda o de la derecha de la página de Moodle. Son uno de los tipos de

módulos que se pueden realizar, y suelen funcionar a través de las versiones de Moodle sin apenas alguna modificación.

- **course:** Este componente de Moodle tiene una importancia obvia, dado que la plataforma se organiza en cursos. Como desarrolladores, podemos desear añadir o modificar formatos de cursos e informes.
- **enrol:** La carpeta *enrol* contiene todos los módulos de matriculación de Moodle. Estos módulos controlan la creación y administración de las matriculaciones a nivel de curso.
- **files:** El componente *files* permite a Moodle incorporar archivos al sistema. Esto incluye subida de archivos, control de acceso, y visualización de ficheros.
- **filter:** El sistema de filtros de Moodle es una facilidad de búsqueda y reemplazo basada en expresiones de texto y expresiones regulares. Los filtros encuentran coincidencias y modifican la página antes de ser mostrada. Por ejemplo, existe un filtro que soporta auto-conversión del lenguaje de marcado TEX a gráficos de ecuaciones.
- **lang:** La carpeta *lang* almacena las cadenas de idioma del núcleo del sistema. Esta es la base del soporte de localización e idioma de Moodle.
- **lib:** La carpeta *lib* almacena las librerías de funciones del núcleo del sistema. Cuando se desarrollen módulos o se personalice el sistema, se usarán clases y funciones definidas en esta carpeta.
- **mod:** La carpeta *mod* almacena los módulos de actividad como Tarea, Wiki, Lección o Foro. Las actividades educativas son el núcleo de cada curso desarrollado en Moodle. Los módulos de actividad son más difíciles de crear que los bloques y deben diseñarse de manera que instruyan al estudiante.
- **theme:** La carpeta *theme* almacena todos los temas prediseñados de Moodle y cualquier tema personalizado instalado en el sistema. Los temas son una combinación de CSS, HTML y PHP.

#### 3.1.1.6. Código de Moodle

Como PHP es un lenguaje interpretado, el código de Moodle se almacena como ficheros de código fuente en el servidor web. Cuando un fichero en particular es solicitado por el

servidor, el intérprete PHP parsea el código al instante y la salida resultante se envía a través del servidor web. Como se ha mencionado antes, la “M” de Moodle significa “Modular”, y su estructura de directorios refleja esta modularidad, ya que cada carpeta del nivel superior representa un componente de Moodle. Cada módulo de expansión tiene su propia carpeta dentro del directorio del componente.

Desde el punto de vista del usuario final, los módulos o componentes se instalan copiando su carpeta dentro del directorio apropiado en el servidor. Moodle detecta el nuevo módulo la siguiente vez que un administrador accede al sistema, localiza el código SQL del módulo, lo ejecuta y finalmente muestra por pantalla los resultados.

Los procesos de actualización se realizan de la misma manera, haciendo un seguimiento de la versión de la base de datos y actualizando automáticamente si es necesario.

#### **3.1.1.7. Base de datos de Moodle**

La base de datos de Moodle está organizada en aproximadamente doscientas tablas relacionadas. La instalación por defecto añade el prefijo *mdl\_* al nombre de cada una de las tablas. Cada componente principal del sistema normalmente tiene una o más tablas, cada una comenzando con el nombre del componente. Por ejemplo, hay dos tablas relacionadas con el componente *config*: *mdl\_config* y *mdl\_config\_plugins*. Como programadores, tendremos que modificar la base de datos de forma regular. Por tanto es importante para nosotros ser capaces de tratar la base de datos completa como una entidad, copiando y moviendo instancias de una base de datos completa de Moodle para la creación de áreas de pruebas en las que desarrollará nuestro código. Normalmente haremos esto con herramientas de línea de comandos *MySQL*.

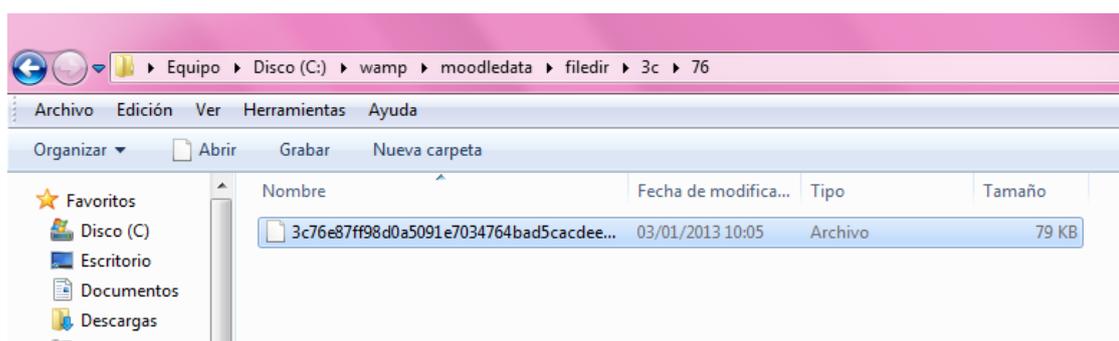
#### **3.1.1.8. Datos de Moodle**

Los ficheros subidos por los usuarios al sistema se almacenan en la localización *moodledata*. En esta carpeta también se almacenan los datos de sesión de los usuarios que se han *logueado* en el sistema si las sesiones basadas en fichero están configuradas.

En los datos de Moodle también se almacenan paquetes de idioma opcionales que pueden ser descargados desde la interfaz de administración. Moodle estructura los datos de esta carpeta bien por usuarios o bien por cursos. Cada curso tiene una carpeta, nombrada con un número entero que representa el identificador único asignado a dicho curso en la base de datos. Podemos determinar fácilmente estos valores navegando en un curso a través de la página de Moodle e inspeccionando los parámetros en la URL. Por ejemplo, examinando esta URL de un curso:

*<http://localhost/moodle/course/view.php?id=4>*

Podemos ver que *id=4*, al final de la cadena, será el identificador de nuestro curso. Si hemos subido ficheros a este curso, existirá una carpeta *moodledata/4*. En esta carpeta, Moodle almacena datos de sus módulos en la subcarpeta *moddata*. Cuando un módulo necesita almacenar ficheros, los almacena dentro de esta carpeta en un directorio con el mismo nombre que el módulo. Por ejemplo, el módulo Wiki tendrá aquí una carpeta llamada *wiki*. Además, Moodle creará una carpeta llamada *backupdata* si se han realizado copias de seguridad de algún curso. Los ficheros que han sido subidos directamente por un usuario usando la interfaz de ficheros del curso se encontrarán en la raíz de esta carpeta. Los usuarios pueden crear también sus propias carpetas y subcarpetas dentro del directorio raíz.



**Ilustración III. 3: Organización de ficheros de moodledata en Moodle 2.3**

A continuación haremos un repaso por todas las actividades, recursos y bloques que componen el paquete estándar de instalación de Moodle.

### 3.1.1.9. Actividades

Las actividades con las que cuenta Moodle por defecto son las que se muestran en la ilustración siguiente. Desde la página de Moodle se pueden descargar otras muchas con distintas funcionalidades, e incluso nosotros mismos podemos desarrollar nuestra actividad personalizada.

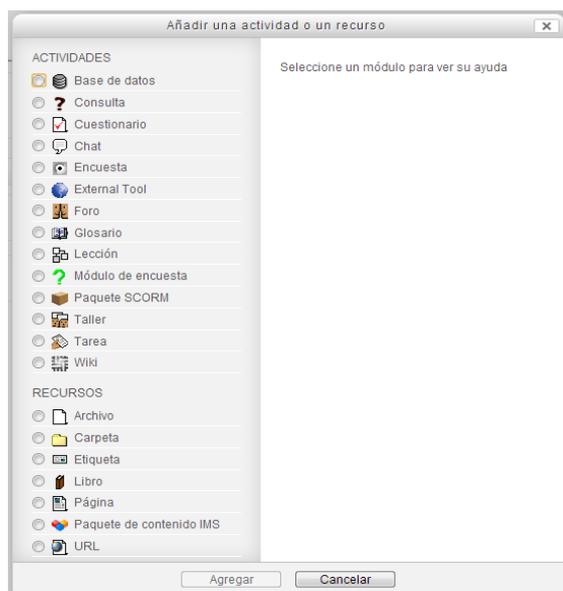


Ilustración III. 4: Actividades y Recursos instaladas por defecto en Moodle 2.3

### 3.1.1.10. Recursos

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

- **Editar una página Web:** La página de texto y, especialmente, la página web son dos recursos muy adecuados para publicar todo tipo de contenidos. En concreto, la creación de página web nos permite añadir materiales curriculares propios o personalizados utilizando el editor integrado de Moodle.
- **Enlazar un archivo o una página Web:** El enlace a una web es útil para añadir contenidos externos al sitio Moodle de forma rápida o para facilitar el acceso y conocimiento de sitios de interés general (buscadores, diccionarios,

instituciones, etc.), mientras que el enlace a un archivo nos permite agregar contenidos con múltiples formatos, comprimidos o no, con la ventaja adicional de que no es obligatorio que estén en el sitio Moodle (aunque requiere la configuración por parte del administrador).

- **Desplegar paquetes de contenido IMS:** IMS es un esqueleto de especificaciones que ayuda a definir variados estándares técnicos, incluyendo materiales de e-learning. La especificación IMS (Content Packaging Specification) hace posible almacenar los contenidos en un formato estándar que puede ser reutilizado en diferentes sistemas sin necesidad de convertir dichos contenidos a otros formatos.

### 3.1.1.11. Bloques

El aspecto del entorno que envuelve un curso en Moodle puede variar en función de las necesidades de alumnos y profesores. Esta personalización se consigue mediante el uso, por parte del profesor del curso, de los denominados bloques de Moodle, que aparecen a la izquierda o la derecha de la pantalla. Existen varios tipos de bloques con funcionalidades muy diversas: informar, controlar, gestionar, etc. Algunos trabajan de forma independiente y otros colaboran entre sí. A continuación se describen los bloques establecidos por defecto en Moodle. A ellos podemos agregar cuanto deseemos, de acuerdo con la funcionalidad que queramos instalar en la plataforma Moodle.

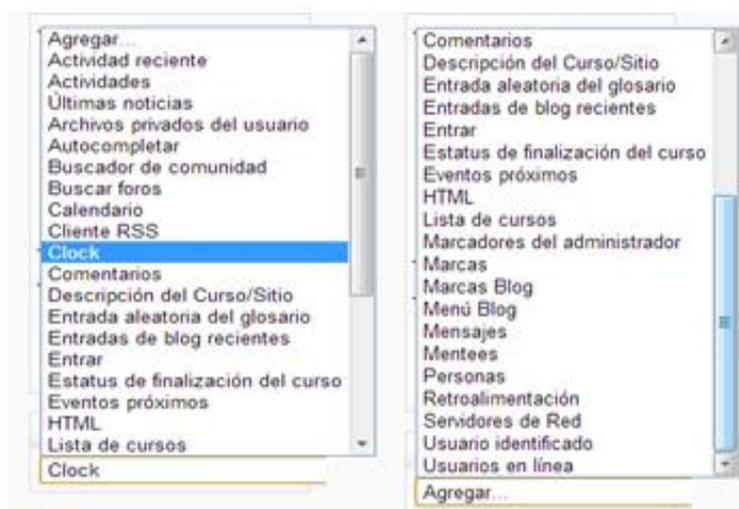


Ilustración III. 5: Bloques Instalados por defecto en Moodle 2.3

### **3.1.1.12. Control de acceso, inicio de sesión y roles de usuario**

Al ser una plataforma tan abierta, Moodle debe ser riguroso con su seguridad. Por ello hace grandes esfuerzos para asegurar que ninguna persona acceda al sistema si no debe hacerlo, y que los usuarios que acceden lo hagan de la manera correcta.

Moodle cuenta con un potente sistema de permisos y control de acceso, cuyo núcleo son las cuentas de usuario. Aunque es posible permitir el acceso a un curso a cualquier visitante sin necesidad de autenticación, a estos usuarios invitados no les está permitido hacer algunas cosas importantes en el sistema. Lo usual es que queramos saber quiénes son nuestros usuarios, y que cada uno tenga su propia cuenta.

Una cuenta de usuario proporciona acceso individual al sistema mediante un nombre de usuario y una contraseña. Estas cuentas se crean utilizando los plugins de autenticación. Las contraseñas en Moodle se codifican con una función hash MD5 para hacerlas ilegibles y difíciles de averiguar.

#### **3.1.1.12.1. Permisos**

Los permisos en Moodle pueden ser asignados dentro de seis contextos: sitio/global, categoría de curso, curso, bloques y actividades, usuario y portada. Existen siete roles integrados: administrador, profesor, profesor no editor, estudiante, creador de cursos, usuario autenticado e invitado, y todos ellos pueden ser asignados en uno o más de los contextos mencionados.

Como desarrolladores, podemos crear capacidades para controlar el acceso a las nuevas funcionalidades que despluguemos en el sistema, y también se puede crear cualquier número de roles personalizados a partir de la lista de unas doscientas capacidades del sistema. A cada capacidad se le puede asignar uno de estos cuatro niveles de acceso: No configurada, permitida, prohibida y prevenida. Cada usuario puede tener múltiples roles que heredan permisos de todos los niveles de contexto aplicables a una petición de acceso realizada por el usuario. La combinación de todo lo anterior proporciona una solución poderosa y flexible para los administradores.

Los roles estándar del sistema son:

- **Administrador:** El administrador del sistema tiene todos los permisos.
- **Creador de cursos:** Puede crear cursos en el sistema y puede estar limitado a una categoría de cursos.
- **Profesor:** Puede administrar un curso, y además puede desarrollar y actualizar su contenido.
- **Profesor no editor:** Puede administrar un curso, pero no puede modificar su estructura.
- **Estudiante:** Puede estar matriculado en un curso.
- **Usuario autenticado:** Todo usuario que haya iniciado sesión en el sistema tiene este rol.
- **Invitado:** Los usuarios no autenticados que tiene permiso de acceso al sistema.

Se pueden crear multitud de roles personalizados utilizando la interfaz de Moodle. Para comprobar y forzar un inicio de sesión dentro de nuestro código, se debe realizar una llamada a *require\_login()*. Esta función permite comprobar si un usuario ha iniciado sesión, y forzar a que lo haga si lo requiere el elemento al que está tratando de acceder. Por ejemplo, podemos crear una actividad específica en un módulo que requiera que el usuario haya iniciado sesión en el sistema. Si no lo ha hecho, será re direccionado a la función de *login* adecuada.

### 3.1.1.12.2. Capacidades

Las capacidades están asociadas con niveles de contexto y son reglas específicas de acceso que pueden ser concedidas a los roles. Ejemplos de capacidades son los siguientes:

- *moodle/site:manageblocks:* Puede administrar bloques en el nivel de contexto sitio.
- *moodle/user:viewdetails:* Puede ver los detalles de un usuario en el nivel de contexto usuario.

➤ *moodle/course:view*: Puede ver un curso en el nivel de contexto curso.

Cuando se desarrollen nuevas capacidades para el código personalizado, se debe considerar cuidadosamente en qué contexto estarán mejor localizadas. Las capacidades normalmente se colocan en el nivel de contexto más bajo en el que puedan funcionar.

### **3.1.1.12.3. Roles**

Los roles son identificadores específicos asociados con todos los contextos. Principalmente se usan para agrupar capacidades de un contexto, de forma que este grupo de capacidades puede ser asignado a los usuarios. Las capacidades se asignan a roles en contextos específicos bien por defecto o mediante una asignación específica.

Son la asignación de roles a nivel de contexto y las capacidades que un rol tiene en ese contexto lo que determina si un usuario puede realizar la acción requerida.

Como desarrolladores de código para Moodle, necesitaremos tener esta funcionalidad en mente, para que de esta manera podamos diseñar y construir el control de acceso que deseamos para nuestras funciones.

### **3.1.1.12.4. Instalación de Moodle**

Ver anexo 1.

### **3.1.1.12.5. Guías de estilo para desarrolladores**

#### **3.1.1.12.5.1. Estilo del código**

#### **3.1.1.12.5.2. Reglas generales**

Este apartado se basa en las “Reglas Generales” que propone Moodle para sus desarrolladores [15]

- Todos los archivos de código deben utilizar la extensión .php.
- Todas las plantillas deben utilizar la extensión .html.
- Todos los archivos de texto deben utilizar el formato de texto Unix (la mayoría de los editores de texto tienen esto como una opción).
- Todas las etiquetas PHP deben ser 'completas', como `<?php ?>`, no 'reducidas', como `<? ?>`.
- Todos los avisos de copyright deben ser mantenidos. Podemos incluir los nuestros si resulta necesario.
- Todos los archivos deben incluir el archivo principal `config.php`.
- Cualquier otro `include/require` debe utilizar una ruta absoluta que comience por `$CFG->dirroot` o `$CFG->libdir`, nunca relativos, ya que estos en algunas ocasiones funcionan de forma extraña en PHP.
- Cada archivo debe comprobar que el usuario está autenticado correctamente, utilizando las funciones `require_login()`, `require_capability()` y `has_capability()`.
- Todos los accesos a la base de datos deben utilizar las funciones definidas en `lib/datalib.php` cuando sea posible, ya que esto permite la compatibilidad con un gran número de bases de datos y prácticamente todo es posible utilizando estas funciones. La inclusión de código SQL queda restringida a funciones específicas de nuestro código (normalmente en el archivo `lib.php`). Además debemos comprobar que funciona en cualquier plataforma y debe estar claramente comentado.
- No se deben crear o utilizar variables globales distintas de las estándar `$CFG`, `$SESSION`, `$THEME`, `$SITE`, `$COURSE` y `$USER`.
- Todas las variables deben ser inicializadas o, al menos, comprobada su existencia utilizando `isset()` o `empty()` antes de ser utilizadas.
- Todas las cadenas deben ser traducibles. Para ello crearemos nuevos textos en los archivos `lang/es` con palabras reducidas en inglés y su traducción completa al español y las recuperaremos en nuestro código utilizando las funciones `get_string()` o `print_string()`.
- Todos los errores deben ser visualizados utilizando la función `print_error()` para maximizar la traducción y ayudar a los usuarios (Automáticamente se enlaza con Moodle Docs).

- Todos los ficheros de ayuda deben ser traducibles. Para ello crearemos nuevos textos en el directorio lang/es/help y los llamaremos utilizando la función helpbutton(). Si necesitamos actualizar un fichero de ayuda:
- La información que llega desde el navegador (enviada con los métodos GET o POST) automáticamente tiene las "magic\_quotes" aplicadas (sin importar la configuración de PHP) por lo que podemos insertarla con total seguridad en la base de datos. El resto de la información (obtenida desde los archivos, o desde la base de datos) debe ser escapada con la función addslashes() antes de insertarla en la base de datos.
- Las acciones de los usuarios deben ser grabadas utilizando la función add\_to\_log(). Estos registros son utilizados para la generación de los "Informes de Actividad" y los registros.
- Al generar enlaces HTML, deben hacerse siempre relativos a la raíz del sitio Moodle, por ejemplo, enlace a \$CFG->wwwroot/mod/blonk/view.php?id=99 en lugar de únicamente view.php?id=99. Gracias a esto el código funcionará aunque sea llamado por un script que se encuentre en otra carpeta diferente.

### **3.1.1.12.5.3. Estilo del código fuente**

Es necesario cumplir las siguientes normas para poder comprender el código fuente de Moodle correctamente.

- El sangrado del texto debe ser siempre de 4 espacios. No debemos utilizar los tabuladores NUNCA.
- Los nombres de las variables tienen que ser siempre fáciles de leer, procurando que sean palabras en minúsculas con significado en inglés. Si realmente necesitamos más de una palabra, debemos ponerlas juntas.
- Debemos utilizar nombres en plural para las matrices de objetos.
- Las constantes tienen que definirse siempre en mayúsculas, y empezar siempre por el nombre del módulo al que pertenecen. Deberían tener las palabras separadas por guiones bajos.

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

- Los nombres de las funciones tienen que ser palabras sencillas en minúsculas y en inglés, y empezar con el nombre del módulo al que pertenecen para evitar conflictos entre módulos. Las palabras deberían separarse por guiones bajos. Los parámetros, si es posible, tendrán valores por defecto. Debemos comprobar que no haya espacio entre el nombre de la función y los paréntesis.
- Los bloques de código siempre deben estar encerrados por llaves (incluso si solo constan de una línea). Moodle utiliza el siguiente estilo: `if ($quiz->attempts) {`

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

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

```
$var = 'texto sin variables';
$var = "con caracteres especiales como nueva línea \n";
$var = 'una cadena muy muy larga con una '.$sola.' variable';
$var = "algo de $texto con $varias variables $dentro";
```

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

```
/**
The description should be first, with asterisks laid out exactly
like this example. If you want to refer to a another function,
do it like this: {@link clean_param()}. Then, add descriptions
for each parameter as follows.
*
@param int $postid The PHP type is followed by the variable name
@param array $scale The PHP type is followed by the variable name
@param array $ratings The PHP type is followed by the variable name
@return mixed
**/
```

```
function forum_get_ratings_mean($postid, $scale, $ratings=NULL) {
if (!$ratings) {
$ratings = array(); // Initialize the empty array
if ($rates = get_records("forum_ratings", "post", $postid)) {
// Process each rating in turn foreach ($rates as $rate) {
....etc
```

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

```
foreach ($objects as $key => $thing) {
process($thing);
}
if ($x == $y) {
$a = $b;
} else if ($x == $z) {
$a = $c;
} else {
$a = $d;}
}
```

- Cuando estemos realizando una COPIA de un objeto, utilizaremos siempre la función *clone()*, originalmente sólo disponible en php5 (en caso contrario simplemente tendremos una referencia al primer objeto). Moodle nos garantiza que este método funcionará también bajo PHP4.

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

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

### 3.2. DOKEOS[33]

Dokeos es un entorno de e-learning y una aplicación de administración de contenidos de cursos y también una herramienta de colaboración. Es software libre y está bajo la licencia GNU GPL, el desarrollo es internacional y colaborativo. También está certificado por la OSI y puede ser usado como un sistema de gestión de contenido (CMS) para educación y educadores. Esta característica para administrar contenidos

incluye distribución de contenidos, calendario, proceso de entrenamiento, chat en texto, audio y video, administración de pruebas y guardado de registros. Hasta el 2007, estaba traducido en 34 idiomas (y varios están completos) y es usado (a septiembre de 2010) por 9900 organizaciones, según reporta el mismo sitio web de la empresa, medido sin filtrado de posibles duplicados.

Los permisos son implementados en base a los roles. Existen 4 roles predefinidos: administrador, profesor, usuario, invitado. No es posible modificar estos roles o crear un nuevo rol. Los usuarios pueden ser organizados dentro de grupos. Una de las principales diferencias con Claroline es que antes de proceder a la instalación de Dokeos debemos crear una base de datos en nuestro servidor MySQL de forma manual.

No se requiere de entrenamiento técnico específico en la plataforma para poder utilizarla, presenta un entorno fácil de aprender y adaptable a las necesidades de las instituciones y organizaciones. Gracias a la implementación de la especificación SCORM, Dokeos proporciona mecanismos para la gestión del aprendizaje, porque permite importar, exportar, generar test sofisticados, etc., También permite mostrar presentaciones en PowerPoint, presenta informes detallados que se pueden exportar a Excel, Business Objects y tiene una herramienta de videoconferencia para sesiones de aprendizaje en línea.

Para la puesta en marcha de la plataforma Dokeos se requiere un Servidor Web HTTP (Recomiendan Apache), con Base de Datos MySQL y desarrollado en lenguaje PHP. Dokeos posee herramientas de enseñanza y seguimiento como: foros, chat, blogs, videoconferencia, intercambio de documentos y agenda. Por otro lado posee herramientas para administración como: importación de usuarios, gestión de las sesiones formativas, es compatible con OpenID y LDAP.

### **3.2.1. Estructura del código de Dokeos [19]**

Dokeos consiste en diferentes piezas de funcionalidad que reciben el nombre de módulos o herramientas: por ejemplo la agenda, los documentos, los enlaces, el foro. El código de cada uno de estos elementos está situado en una carpeta separada del resto.

Lentamente está evolucionando y convirtiéndose en un código más maduro. Las funciones están siendo usadas más y más en el nuevo código, se piensa en encapsulación y en el diseño por capas, no obstante, no se utilizan objetos muy a menudo. Todo el código que puede ser compartido entre las diferentes herramientas se encuentra recopilado en la carpeta *“inc”*. A partir de la versión 1.4 de Claroline/Dokeos, la creación de tus propios módulos es un poco más fácil: solamente hay que usar tres sentencias de inclusión, y obtenemos el encabezado, el pie y las variables globales de Dokeos.

La carpeta *“inc”* abreviatura de incluye (inclusión) es un directorio de propósito general que es utilizado por todos los módulos estándar, y proporciona todas las variables globales de Dokeos, los encabezados y pies estándar, y muchas otras funciones útiles. Está compuesta de ficheros y algunos directorios. En la carpeta *inc/lib* podemos encontrar el código de las librerías, en estos ficheros se proporciona la funcionalidad que es común, como por ejemplo: funciones para la administración de ficheros, consulta de base de datos, funciones generales. Juntos, todos estos ficheros proporcionan una Interface de Programación de Aplicaciones (Application Programming Interface o API).

En el directorio *inc/conf* tenemos los ficheros de configuración para algunas herramientas. Algunos de estos ficheros son tentativos, es decir, aparecen pero no interactúan con ninguna de las herramientas. Todas estas carpetas están actualmente en progreso y evolución.

**NOTA:** La carpeta *“inc”* se llamaba *“include”* en versiones de Dokeos anteriores a la 1.5.

Es recomendable utilizar una plantilla para hacer un nuevo desarrollo. Además, respeta las convenciones de código utilizadas; haciéndolo conseguiremos que el código de Dokeos sea consistente y más legible para todo el mundo.

Mantener el código nuevo tan modular como sea posible. Dividiéndolo en diferentes secciones, usar funciones, comentar a menudo para explicar lo que hace y procurar

dividirlo en piezas lógicas. Utilizar funciones siempre que sea posible, ya que realmente ayudará a mantener la modularidad del código.

### 3.2.2. El API de Dokeos

Hay varias funciones que pueden ser útiles para los desarrolladores. Las funciones de librería se encuentran en la carpeta “*main/inc*”. Antes de empezar la programación, examinar el contenido de esta carpeta, y de las subcarpeta *main/inc/lib* y *main/inc/conf* para que se pueda hacer una idea de lo que Dokeos puede ofrecer desde el principio.

Una de las últimas incorporaciones es el fichero *main.lib.php*, el cual se encuentra en la carpeta *main/inc/lib/* que tiene unas cuantas funciones de carácter general, y el fichero *database.php*, el cual contiene funciones de consulta a la base de datos (y que se encuentran sin testear en este momento). Estos ficheros se están ampliando constantemente. Por ejemplo, hay una función general para mostrar el título de herramientas, hay una función para mostrar la opción [vista de estudiante / vista del administrador del curso], para mostrar una caja de dialogo en html, etc. No hay funciones de base de datos específicas todavía, solo hay algunas de propósito general, pero es un área del desarrollo en el cual estamos trabajando para mejorarla desde ahora. Si se trabaja en Dokeos, y se necesita una consulta que devuelve, por ejemplo, las propiedades de un grupo, se debería añadir una función *getGroupProperties* (con los parámetros que necesitase) en el fichero *main.lib.php* y hacer a posteriori la llamada a la función. Esto representa un poco más de trabajo al principio, pero hará que el código sea más pequeño y más fácil de mantener.

### 3.2.3. Estructura de la base de datos de Dokeos

Dokeos puede ser instalado usando una única base de datos o bien varias. Normalmente la opción de base de datos múltiple es la opción utilizada. Hay dos bases de datos principales, una para mantener la información de carácter general como usuarios, cursos y facultades; y otra para las estadísticas. Por otro lado, cada curso dispone también de su base de datos.

Un usuario puede seguir varios cursos. Un curso puede tener muchos usuarios. Tenemos definida una tabla extra que relaciona usuarios y cursos: *course\_rel\_user*.

### **3.2.4. Convenciones de código propuestas**

#### **3.2.4.1. Lenguaje**

El código, comentarios y las instrucciones de desarrollo deberán de estar escritas en inglés y no en el lenguaje nativo del desarrollador (por ejemplo español o francés). Es una gran idea que los manuales se encuentren disponibles en muchos lenguajes pero el código con los comentarios y variables deben estar en inglés, el francés o el español son más complicados de comprender.

#### **3.2.4.2. Espacios en blanco**

Añadir una línea en blanco entre líneas de código para crear párrafos y ayudar a visualizarlo. Nadie querría leer un libro de miles de páginas y líneas y sin separaciones entre estas, del mismo modo a nadie le gusta leer código escrito de esta manera.

Igualmente, usa espacios alrededor de los operadores y después de la coma, ya que puede ayudar a hacer el código más legible.

```
function something (param a, paramb)
if ( (a && c)==!b)
    a = b + c;
```

#### **3.2.4.3. Indentación**

Utiliza los tabuladores para indentar. Generalmente, un tabulador equivale a cuatro espacios en blanco.

#### **3.2.4.4. Llaves**

Por favor, utilice esta notación

```
if ($wwwww > $jjjjj)
{
c ó digo...
}
```

En vez de esta otra

```
if ($wwwww > $jjjjj) {
c ó digo ...}
```

### 3.2.4.5. Variables y funciones

- Usaremos el guion bajo para separar palabras en vez de poner en mayúsculas la primera letra de cada palabra después de la primera.
- Las variables y las funciones se nombran de este modo:

```
$my_variable
function test_database_api()
```

Y no de este otro:

```
$myVariable
function testDatabaseApi()
```

- Si es posible, intentar limitar el nombre de las variables a no más de tres palabras.
- Intentar nombrar las variables utilizando los nombres en singular en vez de en plural y teniendo en cuenta el contexto.

### 3.2.4.6. Espacios de nombres (Namespaces)

Intentar evitar el uso de variables globales directamente. PHP carece hasta la fecha del concepto de espacios de nombres (namespaces): cada variable usada es global y puede ser vista en el programa entero (excepto en las funciones). No utilice muchas veces la variable `$global_some_var`, mejor utilizar la función `get_global_var()`. De este modo el programa será más fácil de mantener y de resolver errores.

### 3.2.4.7. Constantes

Las constantes deben de aparecer en MAYUSCULAS y si es aplicable UN\_NOMBRE\_LARGO\_IRA\_ASI

### 3.2.4.8. Clases

Los nombres de las clases deben empezar con la primera letra en mayúsculas (Usuario, y no usuario).

### 3.2.4.9. Comentarios

Los comentarios deben de hacer el código más fácil de leer, no hay que explicar lo que es obvio, se asume que el lector dispone de un conocimiento del lenguaje de programación.

- Usar el estilo de Perl como comentarios #
- Siempre usar el estilo del C como comentario (/\* \*/)
- Usar el estilo de C++ cuando el comentario sea muy pequeño (//). Estos comentarios deberían aparecer cerca del código al que se refieren, siempre que sea posible.
- Cada fichero de debe de empezar con el siguiente encabezado, si contribuyes a un fichero, añade tu nombre a la lista de contribuidores de ese fichero.

```
/*
-----
-
Dokeos - elearning and course management software
Copyright (c) 2004 Dokeos S.A.
Copyright (c) 2003 University of Ghent (UGent)
Copyright (c) 2001 Universite catholique de Louvain (UCL)
Copyright (c) Jean Programmer (jean@somewhere.net)
For a full list of contributors, see "credits.txt".
The full license can be read in "license.txt".
This program 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 2
of the License, or (at your option) any later version.
See the GNU General Public License for more details.
Contact: Dokeos, 181 rue Royale, B-1000 Brussels, Belgium,
info@dokeos.com
-----
*/
```

Ilustración III. 6: Comentarios en Dokeos

### **3.3. AMBIENTE DE PRUEBA I**

#### **3.3.1. Descripción de los módulos de prueba**

Los módulos de prueba son escenarios prototipos que ayudan a verificar y obtener datos los cuales nos permiten identificar con claridad que LMS presenta la mayor productividad en la agregación de componentes.

Los módulos que se desarrollarán son implementados en los dos LMS definidos (Moodle y Dokeos). Estos módulos son desarrollados con las características particulares y arquitectura que presenta cada uno de los LMS, es decir aplicando el patrón de diseño que los identifica.

En cada LMS se probara los mismos escenarios para la comparación de los parámetros de productividad definidos posteriormente y se obtendrá los resultados mediante la experiencia en el desarrollo de cada uno de estos módulos de prueba. La mayoría de resultados se los hace midiendo el tiempo empleado en las actividades de desarrollo.

Los distintos módulos prototipos son realizados utilizando el lenguaje de programación PHP y las bondades que presentan los respectivos LMS, Moodle y Dokeos.

**Componente o módulo 1 (mi bloque):** es un bloque desarrollado para los parámetros Acceso a la base de Datos, Seguridad, Codificación, Facilidad de Mantenimiento, Reutilización, Instalación.

**Componente o módulo 2 (Calculadora):** Es un bloque desarrollado para los parámetros, Instalación, Codificación, facilidad de mantenimiento.

**Componente o módulo 3 (ejemplo):** es un módulo actividad que nos servirá para los parámetros Codificación, Instalación, Facilidad de mantenimiento.

### 3.3.2. Desarrollo de los módulos de prueba en Moodle

El desarrollo de nuevos módulos personalizados mediante la utilización de Moodle sigue claramente una estructura bien definida, dividiendo a los componentes o módulos en Bloques, Módulos de Actividad, Recursos, etc.

#### 3.3.2.1. Módulo 1

Lo denominamos *mibloque* (Ver anexo 3) este componente es un bloque que muestra todos los usuarios existentes en la base de datos Moodle. Dentro de este módulo especificaremos las funciones y métodos que nos permitieron acceder a la base de datos y la manipulación de los datos.

Para lo cual se crea una nueva carpeta con el nombre de *mibloque* dentro de la carpeta de bloques de Moodle en la siguiente ubicación *C:\wamp\www\moodle\blocks* dentro de la cual se crean los archivos:

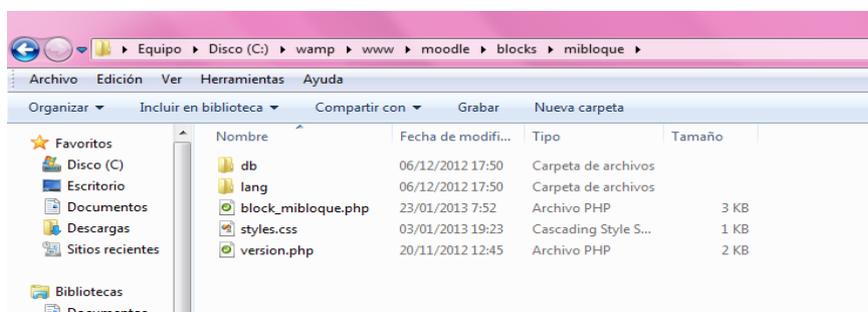


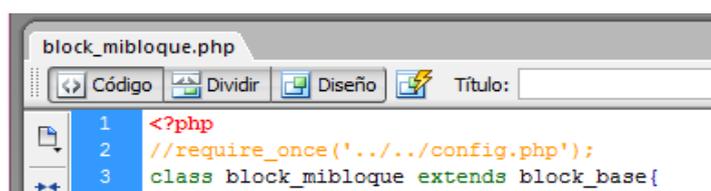
Ilustración III. 7: Archivos y carpetas creadas para el Módulo1

- *block\_mibloque.php*: que es el archivo donde se programó la estructura del componente. (Ver anexo 3)
- *versión.php*: el cuál especifica la versión actual y la versión requerida para la instalación del componente. (Ver anexo 3)
- *styles.css*: es la hoja de estilo con la que dará la apariencia al bloque desarrollado. (Ver anexo 3)

Y las siguientes carpetas:

- La carpeta **mibloque/lang** en la cual se encuentra la carpeta en dentro de la misma está el archivo:
  - **block\_mibloque.php** en el cual se encuentran las cadenas del módulo en este caso del bloque traducidas al idioma inglés y así se crean carpetas de acuerdo a los idiomas en los que se desea mostrar el componente. (Ver anexo 3)
  
- La carpeta **mibloque/db** dentro de la carpeta la cual contiene los archivos:
  - **upgrade.php** la cual describe las acciones a realizar en el momento de una actualización de versión de Moodle. (Ver anexo 3)
  - **access.php** en el cuál se encuentran las capacidades asignadas a los diferentes roles. (Ver anexo 3)

El archivo **mibloque/block\_mibloque.php** utiliza las variables globales del archivo de configuración de Moodle **config.php**. Y con la estructura que requiere el bloque se lo define en la clase heredada de la clase **block\_base**, con la línea de código:



```
block_mibloque.php
Código  Dividir  Diseño  Título:
1  <?php
2  //require_once('../..../config.php');
3  class block_mibloque extends block_base{
```

Ilustración III. 8: Módulo1- código1

Dentro de la clase **class block\_mibloque** están las funciones básicas del bloque:

- **function init():** Es la función esencial en la creación de bloques y su propósito es establecer las dos variables miembro de clase que figuran en su interior. En la cual

- *\$this->title = get\_string('pluginname','block\_mibloque');*: Representa el título de la cabecera del bloque.
- *\$this->version = 2011042200;*: Es la versión de nuestro bloque.

```
class block_mibloque extends block_base{
    function init(){
        $this->title = get_string('pluginname','block_mibloque');
        $this->version = 2011042200;
    }
}
```

Ilustración III. 9: Módulo1-código2

- *function get\_content()*: Es la función donde se programó el contenido del bloque en si aquí se utiliza las variables globales \$USER, \$DB, con la cuales se accede a la base de datos Moodle mediante la función *get\_records\_sql()* para extraer los datos de la tabla *md\_user* de Moodle para lo cual se envían como parámetros la sentencia SQL como se muestra en la ilustración21. La variable global \$OUTPUT nos permite la salida de la imagen del usuario.

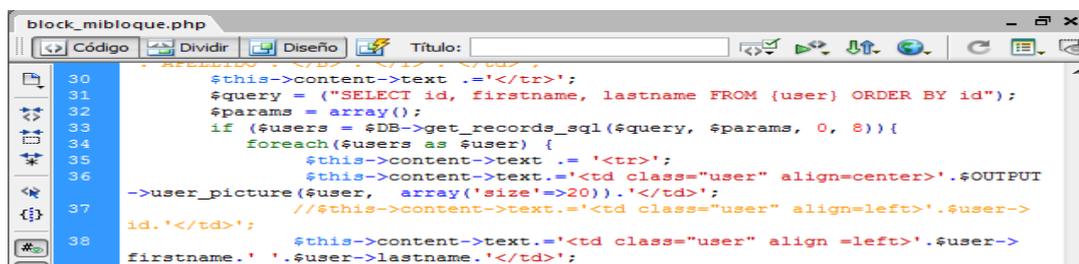


Ilustración III. 10: Módulo1-código3

- *function instance\_allow\_config()*: Es la función de configuración de la instancia.
- *function preferred\_width()*: Para decirle a Moodle ancho preferido de nuestro bloque.

Para su instalación es necesario acceder al sitio Moodle, luego de haber colocado la carpeta donde está instalado Moodle, dentro de la carpeta **blocks** e ingresar a la plataforma como administradores y en la pestaña de Administración del Sitio en la pestaña de Notificaciones nos muestra la pantalla en la cual actualizamos los nuevos

componentes a instalarse. Aparecerá una pantalla como la que se presenta a continuación:

**Número de extensiones que requieren su atención: 6**

Mostrar la lista completa de extensiones (plugins) instalados

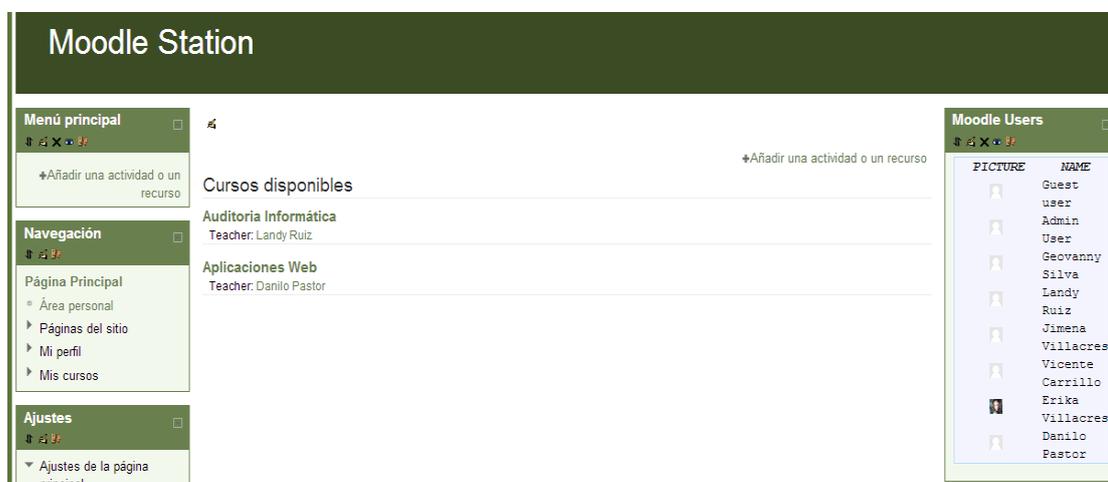
Nombre de la extensión	Directorio	Origen	Versión actual	Nueva versión	Requiere	Estado
<b>Módulos de actividad</b>						
Ejemplo	/mod/ejemplo	Contribución		2010032200	Moodle 2010031900	A instalar
<b>Bloques</b>						
Calculadora	/blocks/calculadora	Contribución	2011061700	2011061700		Instaladas
Configurable Reports	/blocks/configurable_reports	Contribución	2011040103	2011040103	Moodle 2010112400	Instaladas
Moodle Users	/blocks/mibloque	Contribución	2011061700	2011061700		Instaladas
Simple Clock	/blocks/simple_clock	Contribución	2011051300	2011051300		Instaladas
<b>Informes del libro de calificaciones</b>						
[pluginname_gradereport_newreport]	/grade/report/newreport	Contribución	2012061700	2012061700	Moodle 2012061700	Instaladas

Recargar

[Actualizar base de datos Moodle ahora](#)

**Ilustración III. 11: Instalación Módulo 1**

Instalado tendremos que activar la opción de edición de bloques y agregar el bloque que tiene por nombre Moodle Users. Y se nos presentará en los lados izquierdos o derechos de la pantalla de Moodle como se muestra en la siguiente pantalla:



**Ilustración III. 12: Modulo1 instalado**

### 3.3.2.2. Módulo 2

Lo denominamos *calculadora* (Ver anexo 4) este módulo es un bloque, como su nombre lo indica es una calculadora elemental la cual realiza la operaciones básicas que son suma, resta multiplicación y división de dos cantidades ingresadas por el usuario.

Para lo cual crearemos una nueva carpeta con el nombre de *calculadora* dentro de la carpeta de bloques de Moodle en la siguiente ubicación *C:\wamp\www\moodle\blocks* dentro de la cual se crearán los archivos:

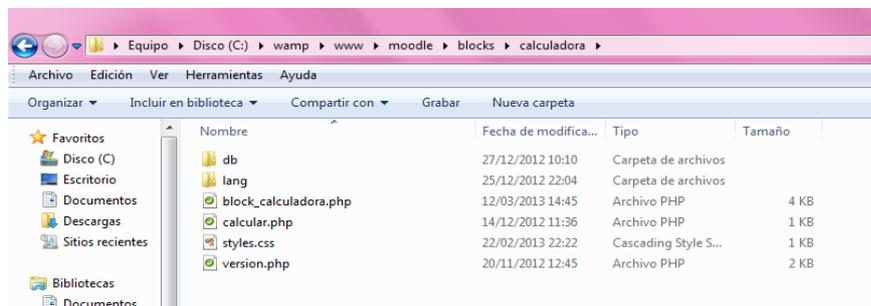


Ilustración III. 13: Archivos creados para Módulo2

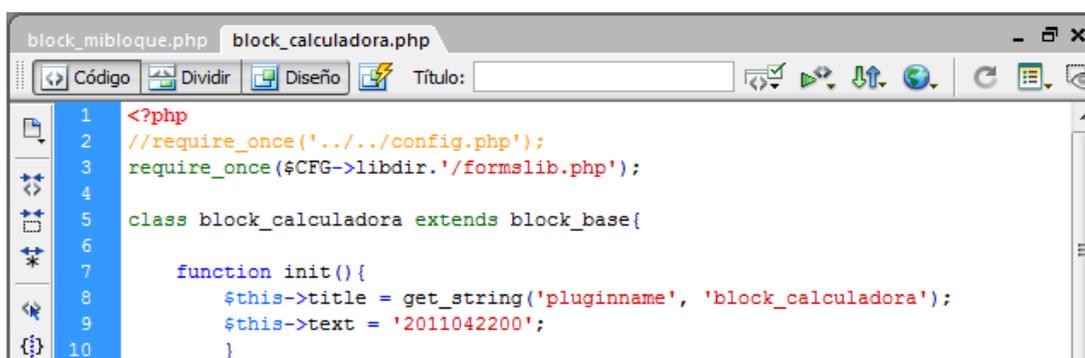
- *block\_calculadora.php*: que es el archivo donde se programará la estructura del componente. (Ver anexo 4)
- *versión.php*: el cuál especifica la versión actual y la versión requerida para la instalación del componente. (Ver anexo 4)
- *styles.css*: es la hoja de estilo con la que dará la apariencia al bloque desarrollado. (Ver anexo 4)

Y las siguientes carpetas:

- La carpeta *calculadora/lang* en la cual se encuentra la carpeta en dentro de la misma está el archivo:
  - *block\_calculadora.php* en el cual se encuentran las cadenas del módulo en este caso del bloque traducidas al idioma en nuestro caso el inglés y así se crearán carpetas de acuerdo a los idiomas en los que se desea mostrar el componente. (Ver anexo 4)
- La carpeta *calculadora/db* dentro de la carpeta la cual contiene el archivo:
  - *upgrade.php* la cual describe las acciones a realizar en el momento de una actualización de versión de Moodle. (Ver anexo 4)

- **access.php** en el cuál se encuentran las capacidades asignadas a los diferentes roles. (Ver anexo 4)

El archivo **mibloque/block\_calculadora.php** se detalla el código de la calculadora. Y se requiere el archivo **formslib.php** para creación de formularios. Con la estructura que requiere el bloque se lo define en la clase heredada de la clase **block\_base**, con las líneas de código siguientes:



```
1 <?php
2 //require_once('../../config.php');
3 require_once($CFG->libdir.'/formslib.php');
4
5 class block_calculadora extends block_base{
6
7     function init(){
8         $this->title = get_string('pluginname', 'block_calculadora');
9         $this->text = '2011042200';
10    }
```

Ilustración III. 14: Módulo 2- Código1

Dentro de la clase **class block\_calculadora** están las funciones básicas del bloque: (Ver anexo 4).

- **function init():** Es la función esencial en la creación de bloques y su propósito es establecer las dos variables miembro de clase que figuran en su interior. En la cual
  - **\$this->title = get\_string('pluginname','block\_calculadora):** Representa el título de la cabecera del bloque.
  - **\$this->version = 2011042200:** Es la versión de nuestro bloque.

```
class block_calculadora extends block_base{
    function init(){
        $this->title = get_string('pluginname', 'block_calculadora');
        $this->version = '2011042200';
    }
}
```

Ilustración III. 15: Módulo2-código2

- **function get\_content():** Es la función donde se programará el contenido del bloque, la calculadora se mostrará en un formulario.(Ver Anexo3).

- *function instance\_allow\_config()*: Es la función de configuración de la instancia.
- *function preferred\_width()*: Para decirle a Moodle ancho preferido de nuestro bloque.

Para su instalación es necesario acceder a nuestro sitio Moodle, luego de haber colocado la carpeta de nuestro bloque *calculadora* dentro de la carpeta *blocks* de nuestro directorio en donde tenemos instalado Moodle e ingresar a la plataforma como administradores y en la pestaña de administración del sitio en la pestaña de Notificaciones nos muestra la pantalla en la cual se actualizan los nuevos componentes a instalarse.

Aparecerá una pantalla como la que se presenta a continuación:

**Número de extensiones que requieren su atención: 6**

Mostrar la lista completa de extensiones (plugins) instalados

Nombre de la extensión	Directorio	Origen	Versión actual	Nueva versión	Requiere	Estado
<b>Módulos de actividad</b>						
 Ejemplo	/mod/ejemplo	Contribución		2010032200	Moodle 2010031900	A instalar
<b>Bloques</b>						
Calculadora	/blocks/calculadora	Contribución	2011061700	2011061700		Instaladas
Configurable Reports	/blocks/configurable_reports	Contribución	2011040103	2011040103	Moodle 2010112400	Instaladas
Moodle Users	/blocks/mibloque	Contribución	2011061700	2011061700		Instaladas
Simple Clock	/blocks/simple_clock	Contribución	2011051300	2011051300		Instaladas
<b>Informes del libro de calificaciones</b>						
[pluginname.gradereport_newreport]	/grade/report/newreport	Contribución	2012061700	2012061700	Moodle 2012061700	Instaladas

 Recargar

[Actualizar base de datos Moodle ahora](#)

**Ilustración III. 16: Instalación Modulo2**

Posteriormente luego de ser instalado se procede a activar la opción de edición de bloques y agregamos el bloque denominado *Calculadora*. Se mostrara de la siguiente manera:

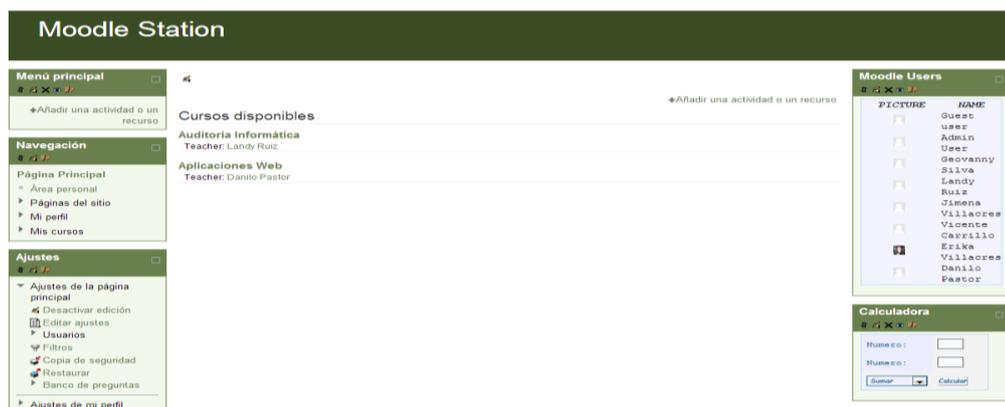


Ilustración III. 17: Módulo2 Instalado

### 3.3.2.3. Módulo 3

El componente *ejemplo* (Ver anexo 5) este módulo es un módulo actividad, el cual representa a una actividad a ser establecida por el docente y realizada por el estudiante. Para lo cual creamos una nueva carpeta con el nombre de *ejemplo* dentro de la carpeta de módulos actividad de Moodle en la siguiente ubicación *C:\wamp\www\moodle\mod* dentro de la cual se crearán los archivos:

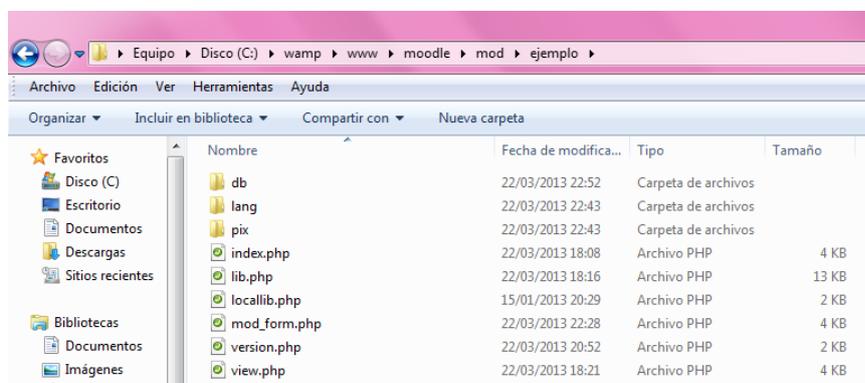


Ilustración III. 18: Archivos y carpetas del modulo3

- *index.php*: lista todas las instancias de una actividad que están presentes en un determinado curso. Normalmente se muestran en formato tabular con una fila por cada instancia de la actividad y con columnas para los campos name, intro y otros campos propios de cada módulo.

- **lib.php**: es la biblioteca de funciones del módulo a desarrollar, en ella se implementará toda (o casi toda) la funcionalidad del módulo. funciones requeridas por Moodle para comunicarse con nuestro módulo.
- **locallib.php**: funciones propias que necesitemos para nuestro módulo.
- **mod\_form.php**: este es uno de los ficheros obligatorios, mediante el cual se configurará cada instancia del módulo que se añada a un curso.
- **versión.php**: Es el archivo donde se encuentra la versión del Moodle para sus posibles actualizaciones.
- **view.php**: En él se define el contenido que se mostrará a cada usuario de la actividad mediante el uso de capacidades o permisos. Página de inicio del módulo.

Se crearon las siguientes carpetas:

- **lang**: carpeta que contiene el archivo de traducción a los diferentes idiomas.
- **db**: carpeta con datos a introducir en la base de datos durante la instalación/actualización del módulo.
  - **access.php**: contiene las capacidades en función a los roles de usuario/permisos del módulo.
  - **install.xml**: esquema de la base de datos en XMLDB para la instalación del módulo.
  - **upgrade.php**: procedimientos de actualización del módulo.
- **pix**: carpeta que contiene el ícono de la actividad que se muestra.

Para su instalación es necesario acceder a nuestro sitio Moodle, luego de haber colocado la carpeta de nuestro módulo **ejemplo** dentro de la carpeta **mod** de nuestro directorio en donde tenemos instalado Moodle e ingresar a la plataforma como administradores y en la pestaña de administración del sitio en la pestaña de Notificaciones nos mostrará la pantalla en la cual deberemos actualizar los nuevos componentes a instalarse.

Aparecerá una pantalla como la que se presenta a continuación:

Número de extensiones que requieren su atención: 6

Mostrar la lista completa de extensiones (plugins) instalados

Nombre de la extensión	Directorio	Origen	Versión actual	Nueva versión	Requiere	Estado
<b>Módulos de actividad</b>						
Ejemplo	/mod/ejemplo	Contribución		2010032200	Moodle 2010031900	A instalar
<b>Bloques</b>						
Calculadora	/blocks/calculadora	Contribución	2011061700	2011061700		Instaladas
Configurable Reports	/blocks/configurable_reports	Contribución	2011040103	2011040103	Moodle 2010112400	Instaladas
Moodle Users	/blocks/mibloque	Contribución	2011061700	2011061700		Instaladas
Simple Clock	/blocks/simple_clock	Contribución	2011051300	2011051300		Instaladas
<b>Informes del libro de calificaciones</b>						
[pluginname.gradereport_newreport]	/grade/report/newreport	Contribución	2012061700	2012061700	Moodle 2012061700	Instaladas

Recargar

[Actualizar base de datos Moodle ahora](#)

Ilustración III. 19: Instalación Módulo3

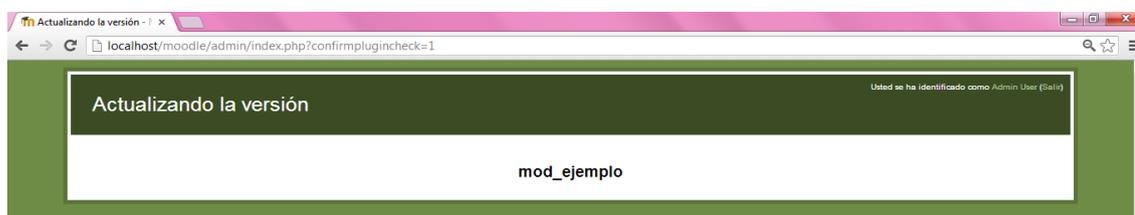


Ilustración III. 20: Actualización en la Base de Datos del Módulo3

Posteriormente luego de ser instalado se procede a activar la opción de edición de bloques y agregamos el bloque denominado *Ejemplo*. Se mostrara de la siguiente manera:

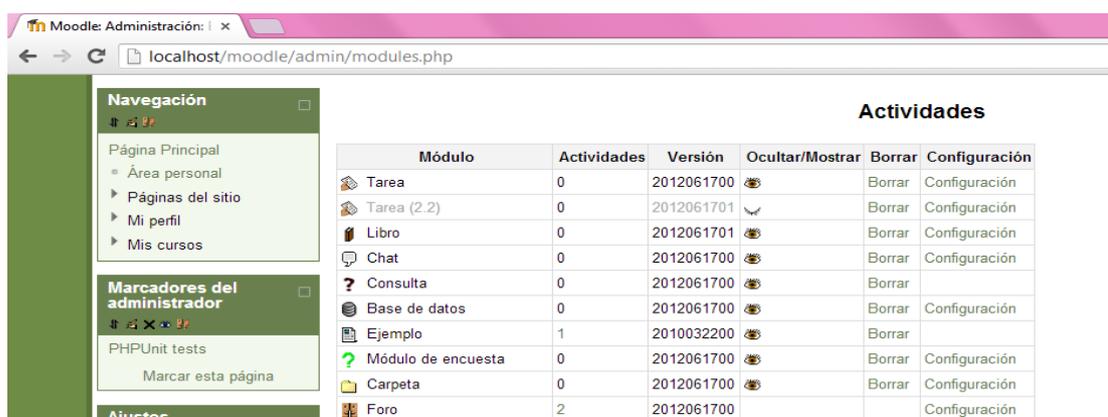


Ilustración III. 21: Módulo 3 Instalado en los Módulos Actividad de Moodle

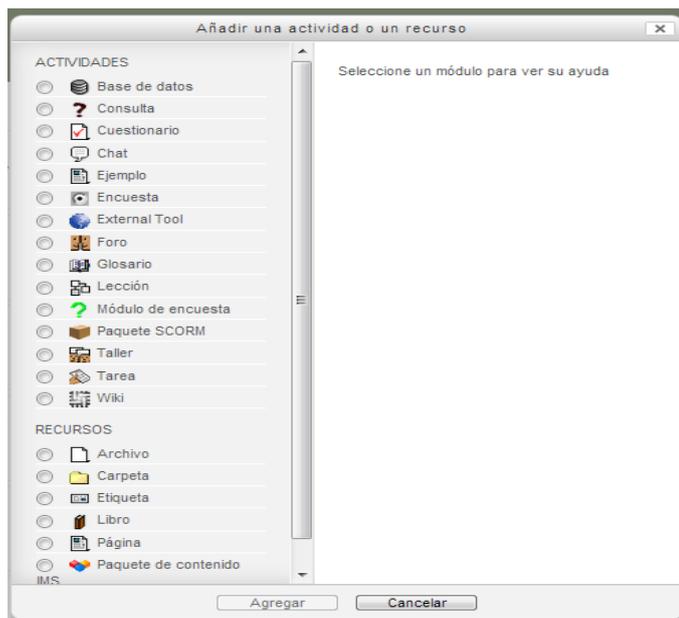


Ilustración III. 22: Agregación del Módulo 3 a un Curso

### 3.4. AMBIENTE DE PRUEBA II: DOKEOS

#### 3.4.1. Desarrollo de los módulos de prueba en Dokeos

##### 3.4.1.1. Módulo 1

#### COMPONENTE DOKEOS\_USER

Para crear un nuevo componente se declara como constante el nombre del módulo en *main/inc/lib/main\_api.lib.php*

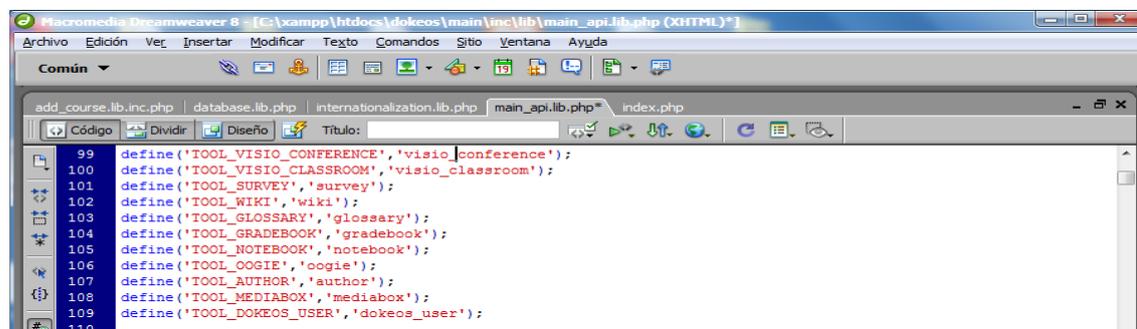


Ilustración III. 23: Módulo 1-Código 1

Crear la carpeta que contendrá el archivo *index.php* (ver anexo 6) dentro de carpeta *main* de Dokeos *C:\xampp\htdocs\dokeos\main*.

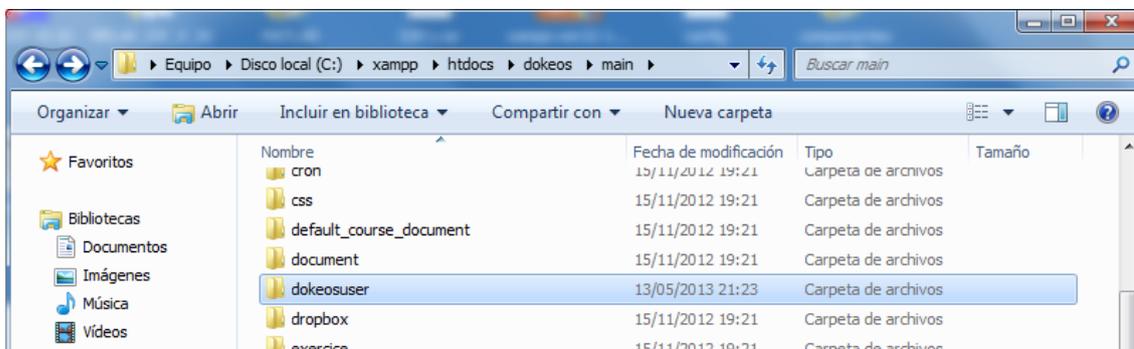


Ilustración III. 24: Creación carpeta Modulo 1

El archivo *dokeosuser/index.php* utiliza un archivo de configuración propio de Dokeos *'../inc/global.inc.php'*.

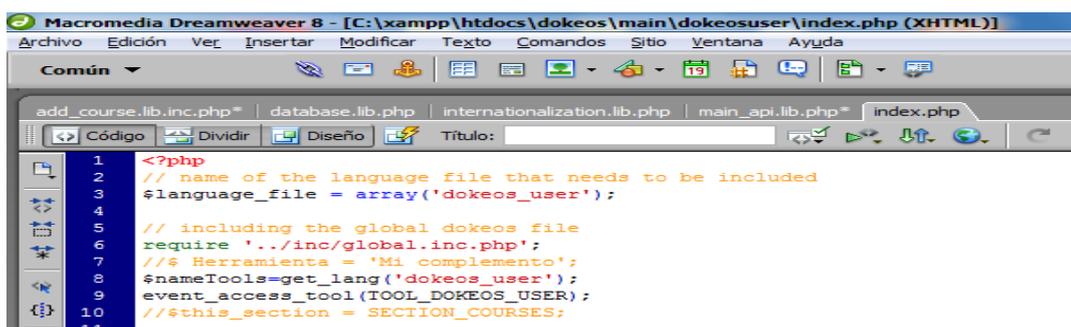


Ilustración III. 25: Modulo 1-Código 2

Se definió el módulo *define('DOKEOS\_USER', true)* y desplegamos nuestra cabecera *Display::display\_tool\_header();*

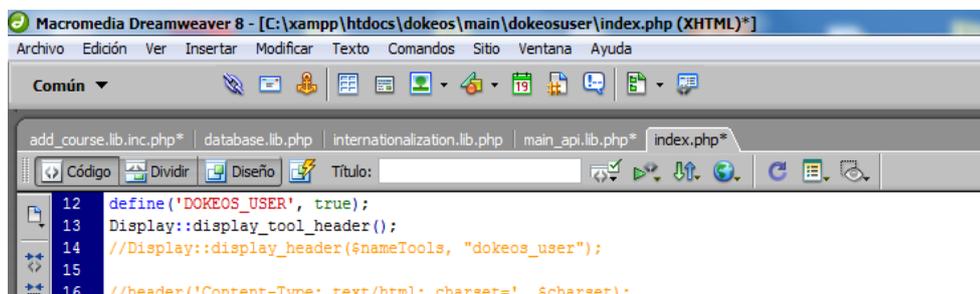


Ilustración III. 26: Modulo 1-Código 3

Ejecutar las sentencias SQL las cuales deberán acceder a nuestra base de datos *\$table\_session\_course\_user=Database::get\_main\_table(TABLE\_MAIN\_SESSION\_COURSE\_USER);* y realizar una búsqueda para traer los datos.

```
$table_session_course_user = Database::get_main_table(TABLE_MAIN_SESSION_COURSE_USER);
$sql_query = "SELECT DISTINCT user.user_id, ".($is_western_name_order ? "user.firstname, user.lastname" : "user.lastname, user.firstname").", user.email,
user.official_code
FROM $table_session_course_user as session_course_user, $table_users as user
WHERE 'course_code' = '".Database::escape_string($course['sysCode'])."' AND session_course_user.id_user = user.user_id ";

if ($session_id != 0) {
    $sql_query .= ' AND id_session = '.$session_id;
}

$sql_query .= $sort_by_first_name ? ' ORDER BY user.firstname, user.lastname' : ' ORDER BY user.lastname, user.firstname';
$rs = Database::query($sql_query, __FILE__, __LINE__);
while ($user = Database::fetch_array($rs, 'ASSOC')) {
    $data[] = $user;
    //$user_infos = Database::get_user_info_from_id($user['user_id']);
    $a_users[$user['user_id']] = $user;
}
}
```

Ilustración III. 27: Modulo 1-Código 4

Añadir la sentencia SQL para la tabla en *main/inc/lib/add\_course.lib.inc.php*, al final del método *update\_Db\_course()*.

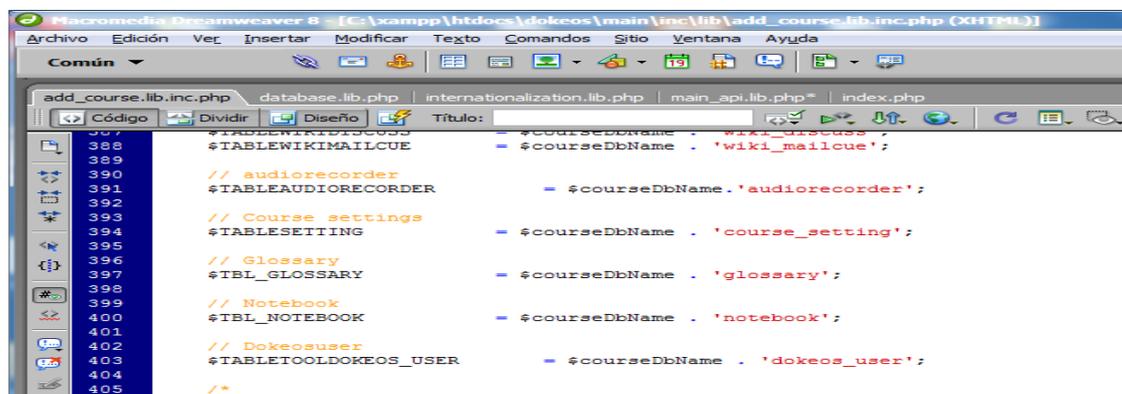
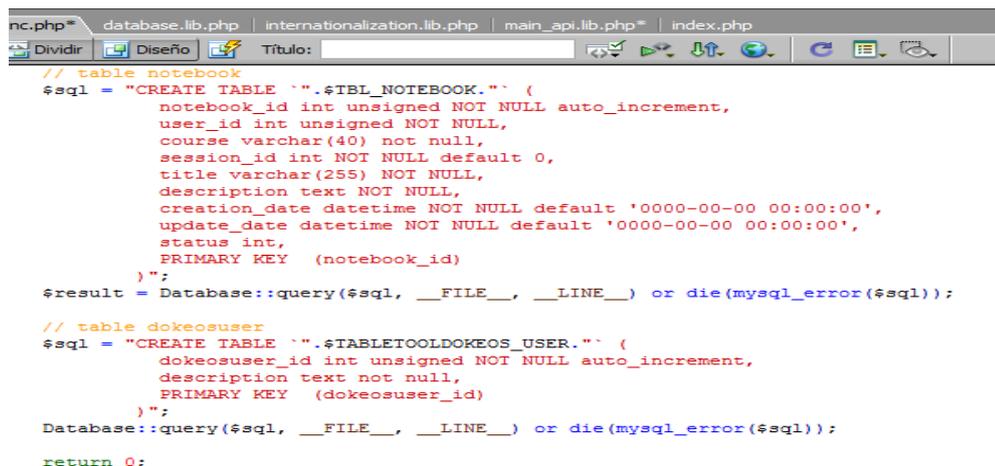


Ilustración III. 28: Actualización nuevo módulo

Crear la tabla que contendrá el nuevo componente mediante sentencias SQL en la base de datos *main*



```
nc.php database.lib.php internationalization.lib.php main_api.lib.php* index.php
Dividir Diseño Título:
// table notebook
$sql = "CREATE TABLE `".$TBL_NOTEBOOK."` (
    notebook_id int unsigned NOT NULL auto_increment,
    user_id int unsigned NOT NULL,
    course varchar(40) not null,
    session_id int NOT NULL default 0,
    title varchar(255) NOT NULL,
    description text NOT NULL,
    creation_date datetime NOT NULL default '0000-00-00 00:00:00',
    update_date datetime NOT NULL default '0000-00-00 00:00:00',
    status int,
    PRIMARY KEY (notebook_id)
)";
$result = Database::query($sql, __FILE__, __LINE__) or die(mysql_error($sql));

// table dokeosuser
$sql = "CREATE TABLE `".$TABLETOOLDOKEOS_USER."` (
    dokeosuser_id int unsigned NOT NULL auto_increment,
    description text not null,
    PRIMARY KEY (dokeosuser_id)
)";
Database::query($sql, __FILE__, __LINE__) or die(mysql_error($sql));

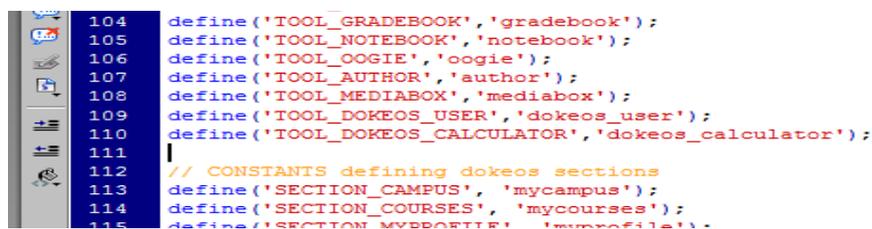
return 0;
```

Ilustración III. 29: Creación tabla para nuevo módulo

### 3.4.1.2. Módulo 2

## COMPONENTE DOKEOS\_CALCULATOR

Para nosotros poder crear un nuevo componente debemos declarar como constante el nombre del módulo en *main/inc/lib/main\_api.lib.php*



```
104 define('TOOL_GRADEBOOK', 'gradebook');
105 define('TOOL_NOTEBOOK', 'notebook');
106 define('TOOL_OOGIE', 'oogie');
107 define('TOOL_AUTHOR', 'author');
108 define('TOOL_MEDIABOX', 'mediabox');
109 define('TOOL_DOKEOS_USER', 'dokeos_user');
110 define('TOOL_DOKEOS_CALCULATOR', 'dokeos_calculator');
111
112 // CONSTANTS defining dokeos sections
113 define('SECTION_CAMPUS', 'mycampus');
114 define('SECTION_COURSES', 'mycourses');
115 define('SECTION_MYDRAFTS', 'mydrafts');
```

Ilustración III. 30: Modulo 2-Código 1

Creamos la carpeta que tendrá nuestro *index.php* (ver anexo 7) en *main* de Dokeos  
*C:\wamp\htdocs\dokeos\main*

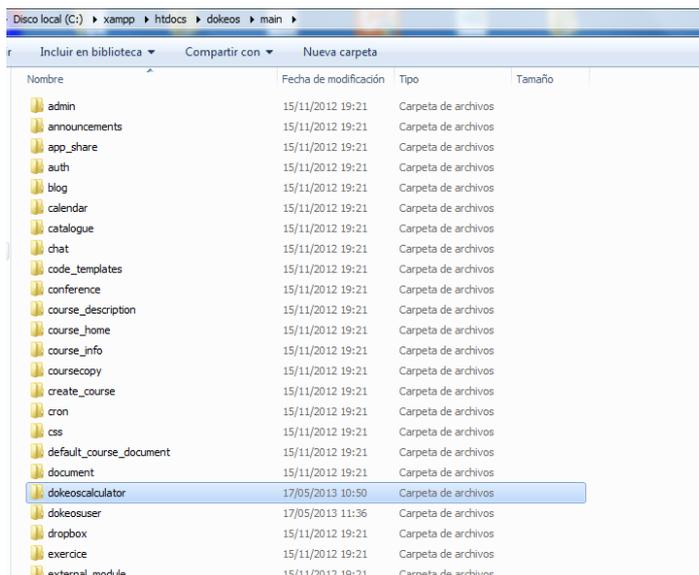


Ilustración III. 31: Creación de carpeta para el nuevo módulo

Configuramos nuestro archivo *index.php* (ver anexo 7) en el que se desarrollara la programación de la calculadora básica.

Configuramos el archivo de lenguaje para q puede ser relacionado con nuestro modulo *\$lenguaje\_file*, declaramos y damos acceso a nuestro módulo *event\_acces\_tool* (*TOOL\_DOKEOS\_USER*), lo definimos y mostramos la cabecera.

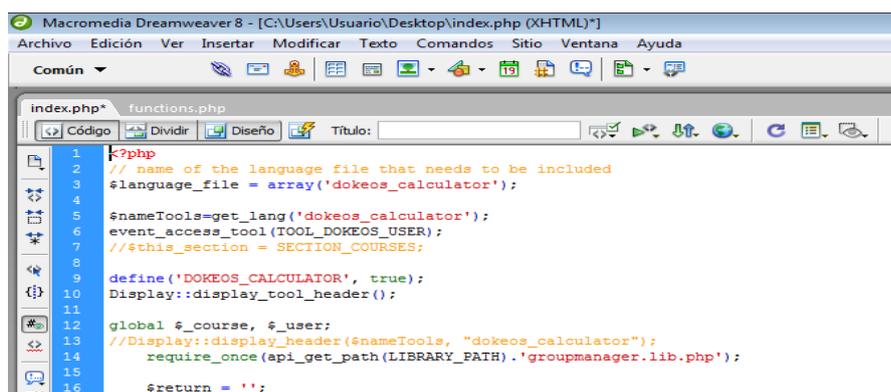


Ilustración III. 32: Modulo 2-Código 2

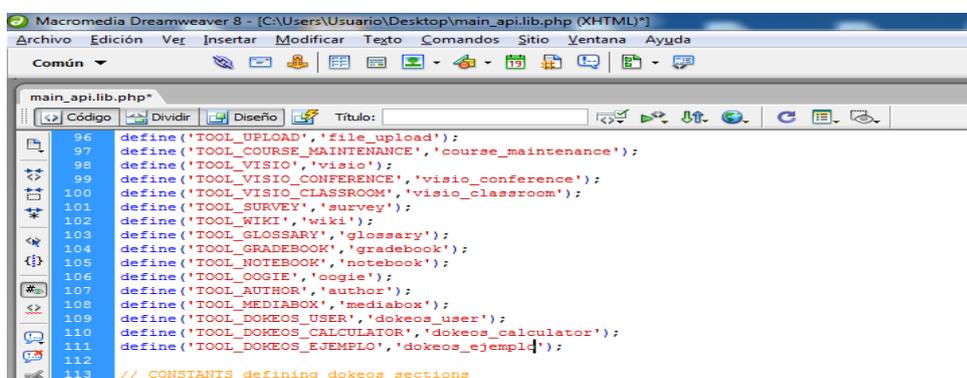
Definimos el nuevo módulo dentro del grupo de interacción para poder usarlo.

```
2367 // Smartblogs (Kevin Van Den Haute :: kevin@develop-it.be)
2368 $sql = "INSERT INTO " . $tbl_course_homepage . " VALUES (NULL, " . TOOL_BLOGS . ", 'blog/blog_admin.php', 'blog_admin.png', " . string2binary(
api_get_setting('course_create_active_tools', 'blogs')) . ", '1', 'squaregrey.gif', 'NO', 'self', 'admin', '0')";
2369 Database::query("INSERT INTO " . $tbl_course_homepage . " VALUES (NULL, " . TOOL_DOKEOS_USER . ", 'dokeosuser/index.php', 'notebook.png', " .
string2binary(api_get_setting('course_create_active_tools', 'dokeos_user')) . ", '0', 'squaregrey.gif', 'NO', 'self', 'interaction', '0')", __FILE__,
__LINE__);
2370 Database::query($sql, __FILE__, __LINE__);
2371 Database::query("INSERT INTO " . $tbl_course_homepage . " VALUES (NULL, " . TOOL_DOKEOS_CALCULATOR .
", 'dokeoscalculator/index.php', 'notebook.png', " . string2binary(api_get_setting('course_create_active_tools', 'dokeos_calculator')) .
", '0', 'squaregrey.gif', 'NO', 'self', 'interaction', '0')", __FILE__, __LINE__);
2372 Database::query($sql, __FILE__, __LINE__);
// end of Smartblogs
```

Ilustración III. 33: Modulo 2-Código 3

### 3.4.1.3. Módulo 3

Para la creación de nuestro tercer módulo o componente, es necesario la definición del mismo en el archivo *main/inc/lib/main\_api.lib.ph*.



```
96 define('TOOL_UPLOAD', 'file_upload');
97 define('TOOL_COURSE_MAINTENANCE', 'course_maintenance');
98 define('TOOL_VISIO', 'visio');
99 define('TOOL_VISIO_CONFERENCE', 'visio_conference');
100 define('TOOL_VISIO_CLASSROOM', 'visio_classroom');
101 define('TOOL_SURVEY', 'survey');
102 define('TOOL_WIKI', 'wiki');
103 define('TOOL_GLOSSARY', 'glossary');
104 define('TOOL_GRADEBOOK', 'gradebook');
105 define('TOOL_NOTEBOOK', 'notebook');
106 define('TOOL_OOGIE', 'oogie');
107 define('TOOL_AUTHOR', 'author');
108 define('TOOL_MEDIABOX', 'mediabox');
109 define('TOOL_DOKEOS_USER', 'dokeos_user');
110 define('TOOL_DOKEOS_CALCULATOR', 'dokeos_calculator');
111 define('TOOL_DOKEOS EjemPlo', 'dokeos EjemPlo');
112
113 // CONSTANTS defining dokeos sections
```

Ilustración III. 34: Definición del componente 3

Crear dentro de la carpeta *main* de la ubicación en la cual se instaló Dokeos, la carpeta del componente. *main/dokeosejemplo*

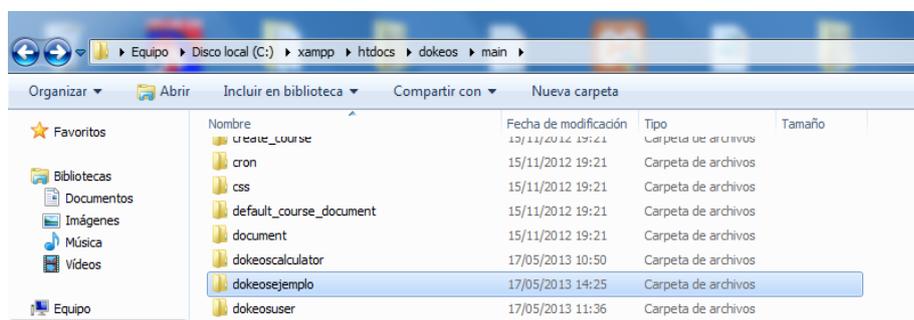
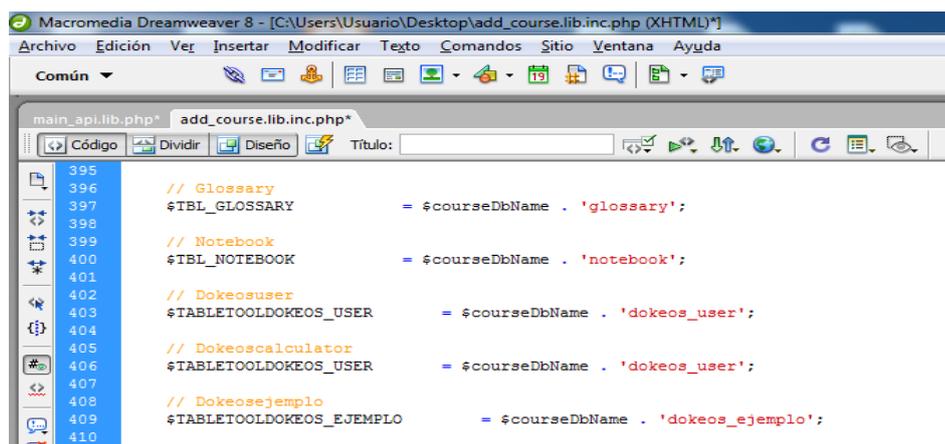


Ilustración III. 35: Creación de la carpeta del módulo 3

Dentro de esta carpeta creamos el archivo *index.php* (ver anexo 8) en el cual se desarrolló la programación de nuestra actividad.

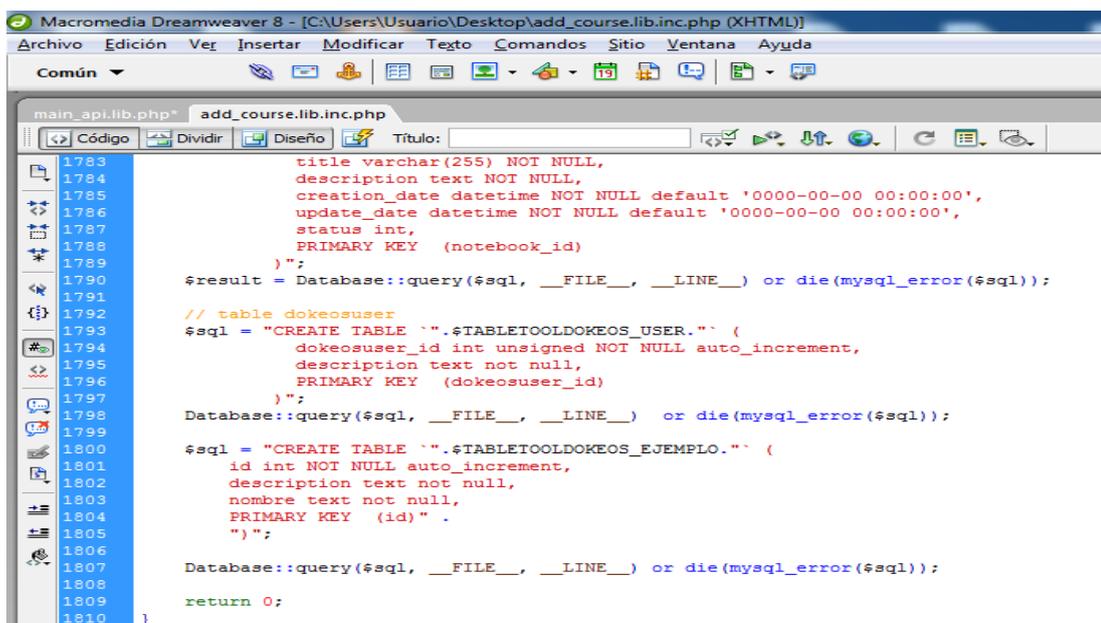
Añadimos la sentencia SQL para definir la tabla del componente en *main/inc/lib/add\_course.lib.inc.php* al finalizar *update\_Db\_course()*.



```
395 // Glossary
396 $TBL_GLOSSARY          = $courseDbName . 'glossary';
397
398 // Notebook
399 $TBL_NOTEBOOK         = $courseDbName . 'notebook';
400
401 // Dokeosuser
402 $TABLETOOLDOKEOS_USER = $courseDbName . 'dokeos_user';
403
404 // Dokeoscalculator
405 $TABLETOOLDOKEOS_USER = $courseDbName . 'dokeos_user';
406
407 // Dokeosejemplo
408 $TABLETOOLDOKEOS_EJEMPLO = $courseDbName . 'dokeos_ejemplo';
409
410
```

Ilustración III. 36: Módulo 3- Código 1

También añadiremos la sentencia SQL en la que se creará la tabla *\$TABLETOOLDOKEOS\_EJEMPLO*.



```
1783     title varchar(255) NOT NULL,
1784     description text NOT NULL,
1785     creation_date datetime NOT NULL default '0000-00-00 00:00:00',
1786     update_date datetime NOT NULL default '0000-00-00 00:00:00',
1787     status int,
1788     PRIMARY KEY (notebook_id)
1789     );
1790 $result = Database::query($sql, __FILE__, __LINE__) or die(mysql_error($sql));
1791
1792 // table dokeosuser
1793 $sql = "CREATE TABLE `".$TABLETOOLDOKEOS_USER."` (
1794     dokeosuser_id int unsigned NOT NULL auto_increment,
1795     description text not null,
1796     PRIMARY KEY (dokeosuser_id)
1797     );
1798 Database::query($sql, __FILE__, __LINE__) or die(mysql_error($sql));
1799
1800 $sql = "CREATE TABLE `".$TABLETOOLDOKEOS_EJEMPLO."` (
1801     id int NOT NULL auto_increment,
1802     description text not null,
1803     nombre text not null,
1804     PRIMARY KEY (id)" .
1805     ")";
1806
1807 Database::query($sql, __FILE__, __LINE__) or die(mysql_error($sql));
1808
1809 return 0;
1810 }
```

Ilustración III. 37: Módulo 3- Código 2

### 3.5. Determinación de parámetros de productividad para la comparación

Para la comparación de los LMS Moodle y Dokeos, en su ámbito de productividad se ha tomado como referencia el manual de FIM – productividad (Fondo para la Investigación y Mejoramiento de la productividad) adaptado al desarrollo de software y los parámetros seleccionados de éste en la investigación CALIDAD SISTÉMICA Y PRODUCTIVIDAD EN EL DESARROLLO DE SISTEMAS DE SOFTWARE [16] desarrollada por Edumilis M. MÉNDEZ, María A. PÉREZ, Anna C. GRIMÁN, Luis E. MENDOZA del Departamento de Procesos y Sistemas de la Universidad Simón Bolívar de Venezuela. Esta investigación se ha seleccionado como guía principal para analizar los parámetros de evaluación de la productividad.

Y es así que se ha considerado una serie de factores, siendo cada índice seleccionado es importante para una mejor determinación de la productividad, se ha dividido en 7 parámetros cada uno con sus sub-parámetros o indicadores para determinar por separado sus potencialidades y debilidades, Se describe a continuación los parámetros de comparación para la determinación del mejor LMS.

Tabla III. I: Parámetros e Indicadores a Valorar

<i>PARAMETROS</i>	<i>INDICADORES</i>
Acceso a Base de Datos	• Soporte para múltiples bases de datos.
	• Manipulación con la base de datos.
	• Desempeño con la base de datos.
Facilidad de Mantenimiento	• Facilidad de manejo de la estructura del LMS (Accesibilidad).
	• Escalabilidad (Actualizaciones).
	• Tiempo requerido en comprender el código (código legible)
	• Especificaciones (versiones-compatibilidad) y estándares

<i>PARAMETROS</i>	<i>INDICADORES</i>
Codificación	• Adaptación de hojas de estilo
	• Estructura de directorios de los componentes del LMS
	• Errores, fiabilidad (ausencia de fallos)
	• Funcionalidad
	• Tiempo de desarrollo
Reutilización	• Herencia de clase base
	• Reutilización de código
	• Tamaño de la aplicación (componentes)
Seguridad	• Utilización de roles y capacidades
	• Integridad de datos
Instalación	• Tiempo de instalación

### 3.5.1. Acceso a Base de Datos

Describe la capacidad del LMS y el análisis de los aspectos que brinda cada herramienta para interactuar con un motor de base de datos y la capacidad de manipulación de los datos.

Tabla III. II: Descripción Parámetro 1: Acceso a Base de Datos

	<i>Indicadores</i>	<i>Descripción</i>
[indicador 1.1]	Soporte para múltiples bases de datos	Describe el comportamiento del LMS ante los distintos motores de base de datos.
[indicador 1.2]	Manipulación con la base de datos	Describe la manipulación de la base de datos y el control de los diferentes elementos que la componen.
[indicador 1.3]	Desempeño con la base de datos	Describe el desempeño del LMS utilizando un motor de base de datos.

### 3.5.2. Facilidad de Mantenimiento

Describe la capacidad que tiene el LMS para brindar facilidad para el mantenimiento de sus componentes tanto en tiempo de desarrollo, estructura, líneas de código, versiones y estándares, etc.

Tabla III. III: Descripción Parámetro 2: Facilidad de Mantenimiento

	<i>Indicadores</i>	<i>Descripción</i>
[indicador 2.1]	Facilidad de manejo de la estructura del LMS (Accesibilidad):	Describe la facilidad de encontrar el directorio o archivo a dar mantenimiento.
[indicador 2.2]	Escalabilidad:	Describe la capacidad del LMS para actualizaciones.
[indicador 2.3]	Tiempo requerido en comprender el código:	Describe el tiempo empleado en comprender el código, su estructura y sintaxis.
[indicador 2.4]	Versiones compatibilidad y Estándares	Describe el cumplimiento a estándares y compatibilidad de código entre las diferentes versiones de los LMS estudiados.

### 3.5.3. Codificación

Describe el nivel de complejidad en la codificación y desarrollados en los componentes en cada LMS.

Tabla III. IV: Descripción Parámetro 3: Codificación

	<i>Indicadores</i>	<i>Descripción</i>
[indicador 3.1]	Adaptación de hojas de estilo	Describe la capacidad de la herramienta para adaptarse a hojas de estilo que personalizaran los componentes.

	<i>Indicadores</i>	<i>Descripción</i>
[indicador 3.2]	Estructura de directorios de los componentes del LMS	Describe la facilidad de la ubicación de los diferentes archivos creados de acuerdo a la estructura de cada LMS.
[indicador 3.3]	Facilidad en la emisión de errores	La facilidad de emisión de errores obtenidos en el desarrollo, instalación y pruebas de los componentes.
[indicador 3.3]	Funcionalidad	La capacidad del LMS de proveer funciones que cumplan con las necesidades específicas.
[indicador 3.4]	Tiempo de desarrollo	El tiempo empleado en el desarrollo de los componentes en las herramientas LMS.

### 3.5.4. Reutilización

Describe la capacidad que brindan las herramientas LMS en reutilizar el código ya programado para los componentes personalizados desarrollados.

Tabla III. V: Descripción Parámetro 4: Reutilización

	<i>Indicadores</i>	<i>Descripción</i>
[indicador 4.1]	Reutilización de código	La capacidad de reutilizar el código de los componentes desarrollados para ser utilizados en la creación de nuevos componentes LMS.
[indicador 4.2]	Herencia Clase Base	La capacidad de heredar clases hijas desde su clase base
[indicador 4.3]	Tamaño de la aplicación	Describe el tamaño en bits que posee el componente desarrollado.

### 3.5.5. Seguridad

Parámetro que mide la capacidad del LMS para brindar componentes y datos íntegros y adecuados.

Tabla III. VI: Descripción Parámetro 5: Seguridad

	<i>Indicadores</i>	<i>Descripción</i>
[indicador 5.1]	Utilización de roles y capacidades	Describe la capacidad del LMS de desarrollar componentes de acuerdo a los roles que desempeñan los diferentes usuarios de las plataformas.
[indicador 5.2]	Utilización de roles y capacidades: Integridad de Datos	Describe la habilidad de resistir a los ataques contra su seguridad.

### 3.5.6. Instalación

Este parámetro indica la facilidad de instalación y su tiempo empleado.

Tabla III. VII: Descripción Parámetro 6: Instalación

	<i>Indicadores</i>	<i>Descripción</i>
[indicador 6.1]	Tiempo de Instalación	Tiempo empleado en instalar cada una de las herramientas LMS.

### 3.6. Análisis Comparativo

En cada LMS se probará los mismos escenarios para la comparación de los parámetros de productividad definidos con anterioridad y se obtendrán los resultados mediante la experiencia en el desarrollo de cada uno de estos módulos de prueba. La mayoría de resultados se los hace midiendo el tiempo empleado en las actividades de desarrollo.

Los resultados de los indicadores con sus respectivos índices se realizan un cuadro comparativo de los LMS Moodle y Dokeos, cuyas pruebas de desarrollo fueron realizadas bajo los mismos escenarios.

La calificación para cada parámetro se determinara de acuerdo a la escala que se mostrará a continuación, lo cual nos permitirá determinar el LMS más productivo al momento de agregar componentes.

Tabla III. VIII: Valoración cualitativa y cuantitativa.

Regular	Bueno	Muy Bueno	Excelente
<70%	>=70% y <80%	>=80% y <95%	>=95%

La evaluación para los indicadores es de acuerdo al tiempo para lo cual la valoración variará entre uno y cuatro, y experiencia de desarrollo que variará ente 1 y 20.

Tabla III. IX: Escala de valoración cualitativa y cuantitativa para los indicadores

Valor Cualitativo		Valor Cuantitativo	Valor Representativo
Insuficiente	No Satisfactorio	1	☺
Parcial	Poco Satisfactorio	2	☺☺
Suficiente	Satisfactorio	3	☺☺☺
Excelente	Muy Satisfactorio	4	☺☺☺☺

Para la realización de la comparación se utilizara la siguiente nomenclatura:

*X* = Representa el puntaje obtenido por el LMS Moodle.

*Y* = Representa el puntaje obtenido por el LMS Dokeos.

*W* = Representa el puntaje sobre el cual será evaluado el parámetro.

*Cm* = Representa el puntaje alcanzado de Moodle en el parámetro.

*Cd* = Representa el puntaje alcanzado de Dokeos en el parámetro.

*Ct* = Representa el puntaje por el cual es evaluado el parámetro.

*Pm* = Calificación porcentual obtenida por Moodle.

***Pd*** = Calificación porcentual obtenida por Dokeos.

Las fórmulas que se utilizaran en el proceso del análisis comparativo son las siguientes:

$$Cm = \sum X$$

$$Cd = \sum Y$$

$$Ct = \sum W$$

$$Pm = \left(\frac{Cm}{Ct}\right) * 100\%$$

$$Pd = \left(\frac{Cd}{Ct}\right) * 100\%$$

### 3.6.1. Acceso a Base de Datos

- **Indicador 1.1: Soporte para múltiples Bases de Datos:** En este indicador valorizará tomando en consideración los números de motores de base de datos que soporta los LMS.

Tabla III. X: Valoración del Indicador 1.1 - Soporte para múltiples Bases de Datos

Número de Base de Datos	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
Hasta 1	1	No Satisfactorio
Hasta 2	2	Poco Satisfactorio
Hasta 3	3	Satisfactorio
4 a más	4	Muy Satisfactorio

- **Indicador 1.2: Manipulación con la base de datos:** En este índice se valorizará de acuerdo a la facilidad de comprensión de la estructura, funciones, sentencias de acceso a la base de datos.

Tabla III. XI: Valoración del Indicador 1.2: Manipulación con la Base de Datos

Dificultad	Valoración	
	Valoración Cualitativa	Valoración cualitativa
Alta	1	No Satisfactorio
Moderada	2	Poco Satisfactorio
Media	3	Satisfactorio
Baja	4	Muy Satisfactorio

- **Indicador 1.3: Desempeño con la base de datos:** En este indicador se valorizará de acuerdo al tiempo en la creación de los modelos de los componentes que posteriormente serán las tablas en la base de datos.

Tabla III. XII: Valoración del Indicador 1.3: Desempeño con la Base de Datos

Tiempo en minutos	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
16 a 20	1	No Satisfactorio
11 a 15	2	Poco Satisfactorio
6 a 10	3	Satisfactorio
1 a 5	4	Muy Satisfactorio

### 3.6.1.1. Valoración

Tabla III. XIII: Resultados Parámetro 1: Acceso a Base de Datos

Indicadores	Moodle		Dokeos	
	Valor Cualitativo	Valor Cuantitativo	Valor Cualitativo	Valor Cuantitativo
Soporte a múltiples DBMS	Muy Satisfactorio	4	No Satisfactorio	1

Indicadores	Moodle		Dokeos	
	Valor Cualitativo	Valor Cuantitativo	Valor Cualitativo	Valor Cuantitativo
Manipulación con la BD	Muy Satisfactorio	4	Satisfactorio	3
Desempeño con la BD	Muy Satisfactorio	4	Satisfactorio	3

### 3.6.1.2. Interpretación

**Soporte a Múltiples sistemas de Base de Datos:** El soporte desde el LMS hacia la administración de la información mediante los distintos motores de base de datos es muy importante, ya que Moodle soporta la mayoría de motores de Base de datos como lo son MySQL, PostgreSQL, Microsoft SQL y Oracle, obteniendo 4 puntos equivalentes a Muy Satisfactorio. Dokeos soporta únicamente MySQL como Motor de Base de Datos, por lo que ha obtenido una calificación de 1 puntos que equivale a No Satisfactorio.

**Manipulación con la Base de Datos:** Este indicador permite determinar la eficiencia que presenta el LMS para conectarse con uno o varios motores de base de datos o configurar su acceso, por lo que el tiempo en realizar esta configuración o conexión describe esta característica. Para el desarrollo de los componentes Moodle utiliza variables globales de su archivo de configuración, **\$DB** de acceso a la base de datos directamente mediante consultas y sentencias SQL sin ser necesario escribir el prefijo ml delante del nombre de la tabla de la cual necesitamos los datos. Siendo indispensable la función **require\_login()** en la cual se verifica que el usuario ha iniciado sesión antes de acceder a cualquier curso o actividades, además verifica el acceso a los cursos, actividades ocultas, y de esta manera saber las capacidades que posee el usuario. De esta manera facilita la manipulación de datos y elimina la necesidad de crear funciones de conexión a la base de datos, siendo su dificultad es Baja, por esto que tiene una calificación de 4 puntos equivalentes a Muy Satisfactorio. Mientras tanto Dokeos al

igual que Moodle posee un archivo de configuración que se encuentra en */main/inc/global.inc.php* dentro de la carpeta donde tenemos instalado nuestro Dokeos utiliza variables de sesión al igual que Moodle para verificar si el usuario esta autenticado previa conexión a la base de datos. Haciendo el trabajo más fácil, con la utilización de **Database** para el acceso a la base de datos, sin embargo Dokeos para cada curso crea una base de datos independiente haciendo el trabajo más dificultoso por ello la manipulación de datos fue media por lo cual se le otorga 3 puntos equivalentes a Satisfactorio.

**Desempeño con la Base de datos:** El desarrollo de los componentes en los respectivos LMS, nos permitió determinar el desempeño que posee cada herramienta con la base de datos. Puesto que el tiempo de implementación de los modelos de los componentes en Dokeos estuvo en el rango de 6 a 10 minutos obtuvo una calificación de 3 puntos que equivale a Satisfactorio, no obstante Moodle presenta una mejor manipulación y facilidad en la creación de los modelos de los componentes siendo estos desarrollados en un rango de 1 a 5 minutos obteniendo una calificación de 4 puntos que equivale a Muy Satisfactorio.

### 3.6.1.3. Calificación

#### Cálculo de Porcentajes

$$C_m = \sum X \qquad C_d = \sum Y \qquad C_t = \sum W$$

$$P_m = \left( \frac{C_m}{C_t} \right) * 100\%$$

$$P_d = \left( \frac{C_d}{C_t} \right) * 100\%$$

$$C_m = 4 + 4 + 4 = 12$$

$$C_d = 1 + 3 + 3 = 7$$

$$C_t = 4 + 4 + 4 = 12$$

$$Pm: \left(\frac{12}{12}\right) * 100\% = 100\%$$

$$Pd: \left(\frac{7}{12}\right) * 100\% = 58\%$$

### 3.6.1.4. Representación de Resultados

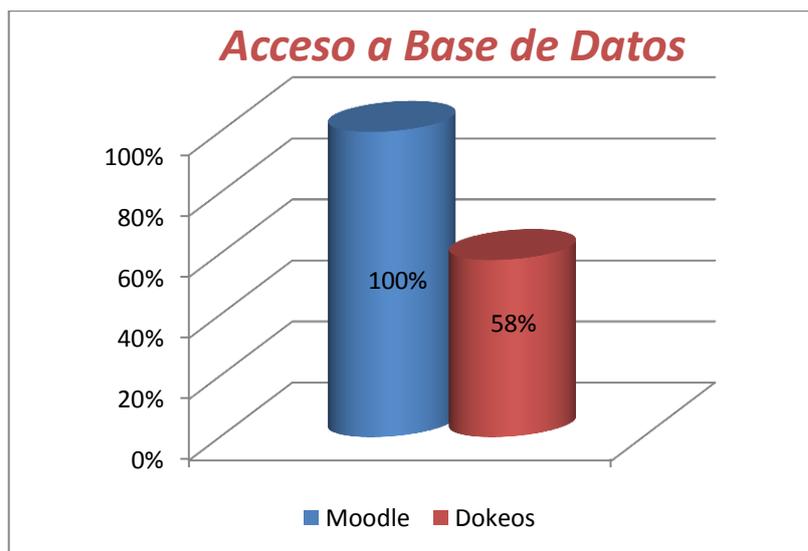


Ilustración III. 38: Representación de Resultados - Parámetro 1: Acceso a Base de Datos

### 3.6.2. Facilidad de Mantenimiento

- **Indicador 2.1: Facilidad de manejo de la estructura del LMS:** En este indicador se valorizará de acuerdo al tiempo empleado en encontrar los archivos que serán modificados para el mantenimiento del componente, en la estructura de directorios del LMS.

Tabla III. XIV: Valoración del Indicador 2.1.: Facilidad de manejo de la estructura del LMS

Valoración	
Implementación/20	Valoración cualitativa
1 a 10	No Satisfactorio
11 a 13	Poco Satisfactorio
15 a 17	Satisfactorio

Valoración	
Implementación/20	Valoración cualitativa
18 a 20	Muy Satisfactorio

- **Indicador 2.2: Escalabilidad:** La Escalabilidad se medirá de acuerdo a la facilidad de actualizaciones.

Tabla III. XV: Valoración del Indicador 2.2: Escalabilidad

Facilidad de Actualizaciones	Valoración	
	Implementación/20	Valoración cualitativa
Alta	1 a 10	No Satisfactorio
Moderada	11 a 13	Poco Satisfactorio
Media	15 a 17	Satisfactorio
Baja	18 a 20	Muy Satisfactorio

- **Indicador 2.3: Tiempo requerido en comprender el código:** Este indicador se valorizará de acuerdo al tiempo empleado en comprender el código de cada componente a actualizar su sintaxis, y estructura en cada herramienta LMS.

Tabla III. XVI: Valoración del Indicador 2.3: Tiempo requerido en comprender el código

Tiempo en minutos	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
41 a 50	1	No Satisfactorio
31 a 40	2	Poco Satisfactorio
21 a 30	3	Satisfactorio
10 a 20	4	Muy Satisfactorio

- **Indicador 2.4: Versiones y estándares:** Este indicador se valorizará en función de la dificultad presentada por la facilidad de cumplimiento a estándares y

compatibilidad de código entre las diferentes versiones para la actualización de componentes en cada una de las herramientas.

Tabla III. XVII: Valoración del Indicador 2.4: Versiones y Estándares

Dificultad de compatibilidad de versiones/Cumplimiento a estándares	Valoración	
	Implementación/20	Valoración cualitativa
Alta	1 a 10	No Satisfactorio
Moderada	11 a 13	Poco Satisfactorio
Media	15 a 17	Satisfactorio
Baja	18 a 20	Muy Satisfactorio

### 3.6.2.1. Valoraciones

Tabla III. XVIII: Resultados Parámetro 2: Facilidad de Actualizaciones

Indicadores	Moodle		Dokeos	
	Valor Cualitativo	Valor Cuantitativo	Valor Cualitativo	Valor Cuantitativo
Facilidad de manejo de la estructura del LMS (Accesibilidad).	Muy Satisfactorio	20	Satisfactorio	17
Escalabilidad (Actualizaciones).	Satisfactorio	17	Satisfactorio	15
Tiempo requerido en comprender el código (código legible)	Satisfactorio	17	Satisfactorio	15
Versiones y estándares	Muy Satisfactorio	20	Satisfactorio	17

### 3.6.2.2. Interpretación

**Facilidad de manejo de la estructura del LMS (Accesibilidad):** Este indicador es fundamental en la productividad de agregación de componentes ya que si no se conoce la estructura ya que sin este conocimiento será mucho más demoroso el encontrar los archivos del componente a actualizar o dar mantenimiento. Moodle por su estructura modular presenta un tiempo menor a 5 minutos por lo que su puntuación en este indicador es 20, mientras que Dokeos posee una estructura un poco más complicada por lo que cada archivo debe ser buscado en diferentes carpetas (Lo explicaremos posteriormente en el indicador Estructura de Directorios de cada LMS), por lo que nos llevó entre 9 a 10 minutos encontrar los archivos del componente a modificar o dar mantenimiento. Por lo que su puntuación es 17 equivalente a Satisfactorio.

**Escalabilidad:** Este indicador es elemental las actualizaciones en Moodle estas se presentan con mayor facilidad por su estructura y por su diseño y la facilidad que presenta la plataforma al presentar actualizaciones al actualizar los archivos de acuerdo a la sintaxis de las versiones y con su archivo `versión.php` y `upgrade.php`. Presentando así una dificultad o facilidad media con 17 puntos equivalentes a Satisfactorio. Por su lado Dokeos en sus actualizaciones dependerá mucho de las líneas de código puramente siendo este trabajo un poco más laborioso por lo que su dificultad es moderada con 15 puntos equivalentes a Satisfactorio.

**Tiempo requerido en comprender el código:** Este indicador representa el tiempo que nos llevó comprender el código de los archivos a dar mantenimiento. En Moodle nos demoramos aproximadamente 25 minutos en comprender su código obteniendo una calificación de 17 puntos equivalentes a Satisfactorio, mientras que en Dokeos nos llevó aproximadamente 40 minutos en comprender su código por ello tiene 15 puntos equivalentes a satisfactorio.

**Versiones y Estándares:** Este indicador es fundamental en Moodle como ya indicamos anteriormente en su estructura modular cada componente posee un archivo ***versión.php*** que posee la versión del componente la misma que será utilizada por el archivo

*upgrade.php* para la actualización automática de la versión del componente si es anterior, antes de esto es necesario conocer la estructura y sintaxis del código ya que a partir de la versión 2.0 difieren mucho, en cumplimiento a estándares Moodle cumple con los principales estándares como lo son SCORM que permite la integración de contenidos por lo que obtiene una calificación de 20 equivalente a Muy Satisfactorio. Por su lado Dokeos no posee archivos que permitan la actualización de versiones y aunque su codificación en sintaxis y estructura no difieren entre sus versiones se debe trabajar únicamente con código y aunque posee el Estándar SCORM por su dificultad al actualizar las versiones del componente obtiene 17 puntos, equivalentes a Satisfactorio.

### 3.6.2.3. Calificación

#### Cálculo de Porcentajes

$$Cm = \sum U \qquad Cd = \sum V \qquad Ct = \sum W$$

$$Pm = \left(\frac{Cm}{Ct}\right) * 100\%$$

$$Pd = \left(\frac{Cd}{Ct}\right) * 100\%$$

$$Cm: 20 + 17 + 17 + 20 = 74$$

$$Cd: 17 + 15 + 15 + 17 = 64$$

$$Ct: 20 + 20 + 20 + 20 = 80$$

$$Pm: \left(\frac{74}{80}\right) * 100\% = 92,5\%$$

$$Pd: \left(\frac{64}{80}\right) * 100\% = 80\%$$

### 3.6.2.4. Representación de Resultados

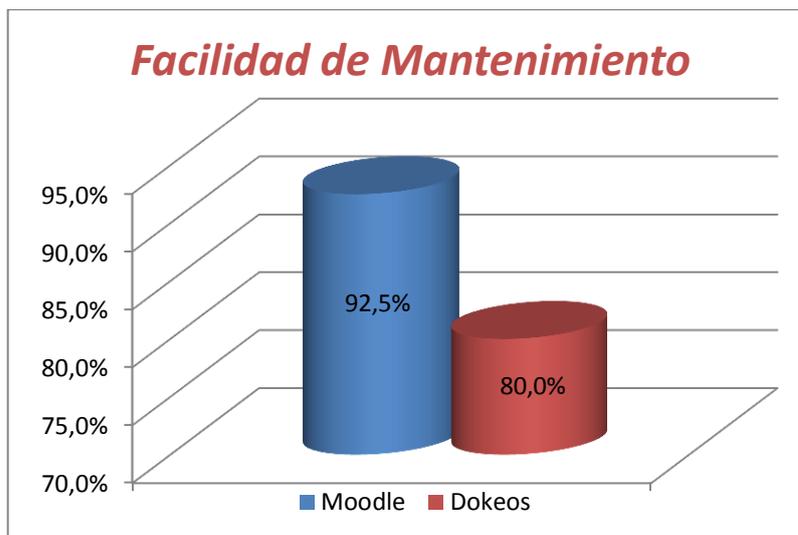


Ilustración III. 39: Representación de Resultados Parámetro2 - Facilidad de Mantenimiento

### 3.6.3. Codificación

- **Indicador 3.1: Adaptación de hojas de estilo:** En este índice se valorizará tomando en consideración la facilidad en la adaptación de los componentes creados con hojas de estilo CSS, permitiendo personalizar los componentes.

Tabla III. XIX: Valoración del Indicador 3.1: Adaptación de hojas de estilo

Facilidad	Valoración	
	Implementación/20	Valoración cualitativa
Baja	1 a 10	No Satisfactorio
Moderada	11 a 13	Poco Satisfactorio
Media	15 a 17	Satisfactorio
Alta	18 a 20	Muy Satisfactorio

- **Indicador 3.2: Estructura de directorios de los componentes del LMS:** Este indicador se medirá de acuerdo a la facilidad de localización de cada uno de los archivos a codificar, su calificación será de 1 a 20.

Tabla III. XX: Valoración del Indicador 3.2: Estructura de los directorios de los componentes del LMS

Facilidad	Valoración	
	Implementación/20	Valoración cualitativa
Baja	1 a 10	No Satisfactorio
Moderada	11 a 13	Poco Satisfactorio
Media	15 a 17	Satisfactorio
Alta	18 a 20	Muy Satisfactorio

- **Indicador 3.3: Facilidad en la emisión de errores:** Este indicador se valorizará de acuerdo a la facilidad de emisión de los errores presentados en el desarrollo de los diferentes componentes en las dos herramientas LMS.

Tabla III. XXI: Valoración del Indicador 3.3: Facilidad den la emisión de errores

Facilidad	Valoración	
	Implementación/20	Valoración cualitativa
Baja	1 a 10	No Satisfactorio
Moderada	11 a 13	Poco Satisfactorio
Media	15 a 17	Satisfactorio
Alta	18 a 20	Muy Satisfactorio

- **Indicador 3.4: Funcionalidad:** Este indicador se valorizará en función de la facilidad y simplicidad de las diferentes funcionalidades presentadas por las herramientas LMS al momento de desarrollar los componentes.

Tabla III. XXII: Valoración del Indicador 3.4: Funcionalidad

Dificultad	Valoración	
	Implementación/20	Valoración cualitativa
Baja	1 a 10	No Satisfactorio
Moderada	15 a 13	Poco Satisfactorio

Valoración		
Media	17 a 16	Satisfactorio
Alta	18 a 20	Muy Satisfactorio

- **Indicador 3.5 Tiempo de Desarrollo:** Este indicador se valorizará en función del tiempo empleado en desarrollar los componentes prototipos dependiendo de la complejidad en su mencionado proceso.

Tabla III. XXIII: Valoración del Indicador 3.5: Tiempo de Desarrollo

Valoración		
Tiempo en minutos	Valoración Cuantitativa	Valoración cualitativa
181 a 240	1	No Satisfactorio
121 a 180	2	Poco Satisfactorio
61 a 120	3	Satisfactorio
10 a 60	4	Muy Satisfactorio

### 3.6.3.1. Valoraciones

Tabla III. XXIV: Resultados Parámetro 3: Codificación

Indicadores	Moodle		Dokeos	
	Valor Cualitativo	Valor Cuantitativo	Valor Cualitativo	Valor Cuantitativo
Adaptación de hojas de Estilo	Muy Satisfactorio	19	Satisfactorio	15
Estructura de los directorios	Muy Satisfactorio	19	Poco Satisfactorio	15
Errores y fiabilidad	Satisfactorio	17	Poco Satisfactorio	15
Funcionalidad	Muy Satisfactorio	19	Satisfactorio	15

Indicadores	Moodle		Dokeos	
	Valor Cualitativo	Valor Cuantitativo	Valor Cualitativo	Valor Cuantitativo
Tiempo de Desarrollo	Muy Satisfactorio	3	Satisfactorio	2

### 3.6.3.2. Interpretación

**Adaptación de hojas de estilo:** La utilización de estilos en los Componentes es muy importante para un diseño personalizado, Moodle permite crear estilos para la adaptación de un solo componente en cada carpeta del mismo siendo este de fácil implementación por lo que obtiene una calificación de 19 que equivale a Muy Satisfactorio. Por otro lado Dokeos, permite la modificación de la hoja de estilo por defecto de la plataforma no permite crear hojas de estilo para cada componente para actualizar o modificar debemos ubicarnos en la dirección *main/css/styles/default.css*, por esto Dokeos obtiene una calificación Satisfactoria de 15 puntos.

**Estructura de los Directorios:** Este indicador es fundamental en el desarrollo de los componentes ya que si no conocemos la estructura de los directorios, existe la posibilidad que no se llegue a obtener los objetivos deseados. Moodle por su estructura modular toda su estructura la tiene organizada en carpetas para cada componente, recurso, etc. Organiza los componentes en módulos actividad (foros, wikis, foros, tareas, etc) *moodle/mod* y bloques *moodle/blocks()*, dentro de cada uno está la carpeta de cada componente *moodle/blocks/nombrecomponente* y dentro de esta los archivos(*nombrecomponente.php, upgrade.php y versión.php*) y carpetas (*db, lang*) correspondientes e indispensables para que los componentes puedan instalarse correctamente. Por esta razón Moodle obtiene 19 puntos por su lado Dokeos presenta una estructura diferente y más dificultosa de entender, todos los nuevos componentes se los coloca dentro de la carpeta *dokeos/main/nombrecomponente* dentro de esta carpeta es indispensable el archivo *index.php* y crear el archivo de lenguaje en *main/lang/[language]/newtool.inc.php* y para la personalización hay que modificar el

archivo de *main/css/[styles]/default.css* como se puede apreciar cada archivo hay que crearlo en una dirección diferente por ello Dokeos tiene 15 equivalente a Satisfactorio.

**Facilidad en la emisión de errores:** En este indicador nos basamos a la facilidad de las herramientas al momento de encontrar un error y comunicarlo o mostrarlo al desarrollador. Moodle al momento de encontrar errores nos envía mensajes de error, en lo que se refiere a sintaxis y acceso a base de datos, en caso de la plataforma en si solo emite una pantalla en blanco. Por ello su calificación es 17 equivalentes a Satisfactorio. Por su parte Dokeos brinda una escasa información sobre errores, y al igual que Moodle emite pantallas en blanco, siendo su calificación 15 equivalentes a Poco Satisfactorio.

**Funcionalidad:** La funcionalidad que cada herramienta ofrece es indispensable en el desarrollo de los componentes. Moodle presenta una alta funcionalidad para el desarrollo de componentes por ello tiene 19 puntos equivalentes a Muy Satisfactorio. Siendo la funcionalidad de Dokeos menor a la de Moodle por ello tiene 15 puntos equivalentes a Satisfactorio.

**Tiempo de Desarrollo:** El tiempo de desarrollo de los componentes dependió de los factores mencionados anteriormente, la creación de los cada uno de los componentes en Moodle tuvo un tiempo de 120 minutos con 3 puntos equivalentes a Satisfactorio. Dokeos por su estructura presento un tiempo de implementación de 150 minutos por lo que si calificación es de 2 puntos equivalentes a Poco Satisfactorio.

### 3.6.3.3. Calificación

#### Cálculo de Porcentajes

$$Cm = \sum U \qquad Cd = \sum V \qquad Ct = \sum W$$

$$Pm = \left( \frac{Cm}{Ct} \right) * 100\%$$

$$Pd = \left( \frac{Cd}{Ct} \right) * 100\%$$

$$Cm: 19 + 19 + 17 + 19 + 3 = 77$$

$$Cd: 15 + 15 + 15 + 15 + 2 = 62$$

$$Ct: 20 + 20 + 20 + 20 + 4 = 84$$

$$Pm: \left(\frac{77}{84}\right) * 100\% = 91,6\%$$

$$Pd: \left(\frac{62}{84}\right) * 100\% = 73,8\%$$

### 3.6.3.4. Representación de Resultados

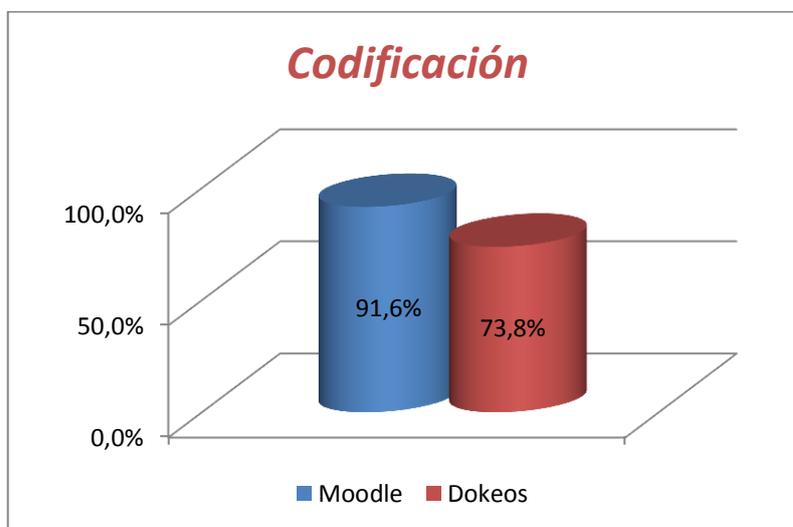


Ilustración III. 40: Representación de Resultados- Parámetro 3: Codificación

### 3.6.4. Reutilización

- **Indicador 4.1: Herencia de Clase Base:** Este indicador se medirá en base a la capacidad de las herramientas LMS de presentar la utilización de clases bases y heredarlas para utilizarlas según las necesidades.

Tabla III. XXV: Valoración del Indicador 4.1: Herencia de Clase Base

Valoración	
Implementación /20	Valoración cualitativa
1 a 10	No Satisfactorio

11 a 13	Poco Satisfactorio
15 a 17	Satisfactorio
18 a 20	Muy Satisfactorio

- **Indicador 4.2: Reutilización de código:** Este indicador se medirá según la capacidad y la facilidad de los LMS de permitir la utilización del código ya desarrollado en otros componentes nuevos o que se van a personalizar.

Tabla III. XXVI: Valoración del Indicador 4.2: Reutilización de Código

Valoración	
Implementación/20	Valoración cualitativa
1 a 10	No Satisfactorio
11 a 13	Poco Satisfactorio
15 a 17	Satisfactorio
18 a 20	Muy Satisfactorio

- **Indicador 4.3: Tamaño de la aplicación:** Este indicador se medirá de acuerdo al tamaño de los componentes finales desarrollados.

Tabla III. XXVII. Valoración del Indicador 4.3: Tamaño de la Aplicación

Tamaño en MB	Valoración	
	Implementación/20	Valoración Cualitativa
> 3	1 a 10	No Satisfactorio
2 a 3	11 a 13	Poco Satisfactorio
1 a 2	15 a 17	Satisfactorio
0 a 1	18 a 20	Muy Satisfactorio

### 3.6.4.1. Valoraciones

Tabla III. XXVIII. Resultados del Parámetro 4: Reutilización

Indicadores	Moodle		Dokeos	
	Valor Cualitativo	Valor Cuantitativo	Valor Cualitativo	Valor Cuantitativo
Herencia de clase Base	Muy Satisfactorio	19	Satisfactorio	13
Reutilización de código	Muy Satisfactorio	19	Poco Satisfactorio	19
Tamaño de la aplicación	Muy Satisfactorio	19	Poco Satisfactorio	17

### 3.6.4.2. Interpretación

**Herencia de Clase Base:** La herencia de clases base son esenciales en Moodle, para crear un bloque nuevo este debe crear una clase que debe heredar de su clase base *block\_base*, para la utilización de formularios también se debe heredar de su clase base *moodleform* y así un sin fin de clases bases existentes. Por lo dicho Moodle trabaja con clases ya existentes que heredan y pueden ser utilizadas en los diferentes componentes de Moodle por ello tiene 19 puntos equivalentes a Muy satisfactorio. Mientras tanto Dokeos posee 13 puntos equivalentes a Satisfactorio ya que no posee herencia de clases no posee clases bases.

**Reutilización de código:** En este indicador ambas plataformas permiten reutilizar el código existe como funciones y librerías, Moodle como ya mencionamos anteriormente permite el uso de clases base, y también permite la reutilización de funciones existentes en los diferentes archivos para lo cual es indispensable incluir los archivos necesarios, al igual que Dokeos, además es posible crear nuestras propias funciones y código el mismo que puede ser utilizado posteriormente, es por ello que ambas herramientas tienen 19 puntos equivalentes a Muy Satisfactorio.

**Tamaño de la Aplicación:** El tamaño de los componentes en Moodle tiene un tamaño menor a 1MB por lo que tiene 19 puntos equivalentes a Muy Satisfactorio y en Dokeos juntando todos sus archivos también su tamaño es inferior a 1 MB ya que a diferencia de Moodle los archivos van en diferentes carpetas por ello tiene 17 puntos equivalentes a Satisfactorio.

### 3.6.4.3. Calificación

#### Cálculo de Porcentajes

$$Cm = \sum U \qquad Cd = \sum V \qquad Ct = \sum W$$

$$Pm = \left(\frac{Cm}{Ct}\right) * 100\% \qquad Pd = \left(\frac{Cd}{Ct}\right) * 100\%$$

$$Cm: 19 + 19 + 19 = 57$$

$$Cd: 13 + 19 + 17 = 49$$

$$Ct: 20 + 20 + 20 = 60$$

$$Pm: \left(\frac{57}{60}\right) * 100\% = 95\% \qquad Pd: \left(\frac{49}{60}\right) * 100\% = 81,6\%$$

### 3.6.4.4. Representación

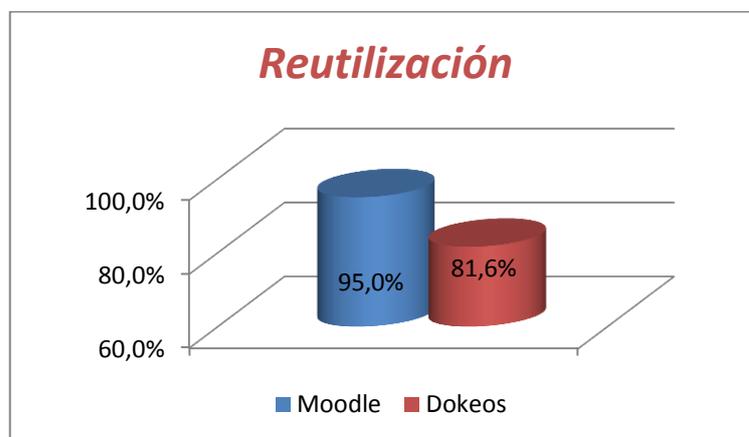


Ilustración III. 41: Representación de resultados - Parámetro 4: Reutilización

### 3.6.5. Seguridad

- **Indicador 5.1: Utilización de roles y capacidades:** Este indicador se medirá de acuerdo a la capacidad y facilidad de los LMS de permitir y utilizar roles y capacidad para mejor seguridad de datos. Su calificación será sobre 20.

Tabla III. XXIX: Valoración del Indicador 5.1: Utilización de roles y Capacidades

Valoración	
Implementación/20	Valoración cualitativa
1 a 10	No Satisfactorio
11 a 13	Poco Satisfactorio
15 a 17	Satisfactorio
18 a 20	Muy Satisfactorio

- **Indicador 5.2: Integridad de datos:** Este indicador se medirá en base a la integridad de datos prestada por ambas herramientas este análisis se lo realizara en base a una calificación sobre 20.

Tabla III. XXX: Valoración del Indicador 5.2: Integridad de Datos

Valoración	
Implementación/20	Valoración cualitativa
1 a 10	No Satisfactorio
11 a 13	Poco Satisfactorio
15 a 17	Satisfactorio
18 a 20	Muy Satisfactorio

### 3.6.5.1. Valoraciones

Tabla III. XXXI: Resultados del Parámetro 5: Seguridad

Indicadores	Moodle		Dokeos	
	Valor Cualitativo	Valor Cuantitativo	Valor Cualitativo	Valor Cuantitativo
Utilización de Roles y Capacidades	Muy Satisfactorio	19	Satisfactorio	19
Integridad de Datos	Muy Satisfactorio	19	Poco Satisfactorio	19

### 3.6.5.2. Interpretación

**Utilización de roles y capacidades:** El uso de los roles y capacidades proporcionan ambas herramientas ya que las dos proporcionan y definen claramente sus roles, y sus capacidades en función de los roles Moodle con la función *has\_capability* controla sus capacidades y Dokeos con preguntar determinada capacidad permite definir acciones por ello tienen 19 puntos equivalentes a Muy Satisfactorio.

**Integridad de Datos:** Ambas Herramientas proporcionan integridad de datos y métodos de encriptación como Md5 para protección de sus contraseñas por ello las dos herramientas tienen 19 puntos equivalentes a Muy Satisfactorio.

### 3.6.5.3. Calificación

$$Cm = \sum U \qquad Cd = \sum V \qquad Ct = \sum W$$

$$Pm = \left(\frac{Cm}{Ct}\right) * 100\%$$

$$Pd = \left(\frac{Cd}{Ct}\right) * 100\%$$

$$Cm: 19 + 19 = 38$$

$$Cd: 19 + 19 = 38$$

$$Ct: 20 + 20 = 40$$

$$Pm: \left(\frac{38}{40}\right) * 100\% = 95\%$$

$$Pd: \left(\frac{38}{40}\right) * 100\% = 95\%$$

### 3.6.5.4. Representación

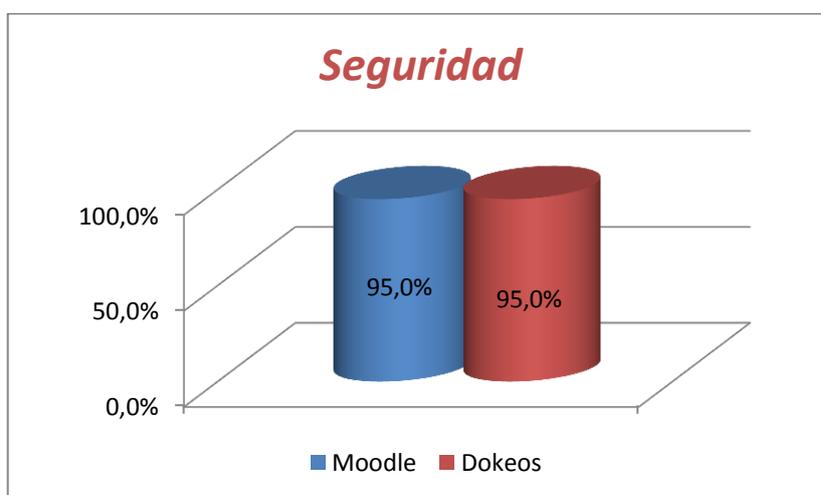


Ilustración III. 42: Representación de resultados - Parámetro 5: Seguridad

### 3.6.6. Instalación

- **Indicador 6.1: Tiempo de Instalación:** Este indicador se valorizará de acuerdo al tiempo empleado en la instalación de los componentes prototipos desarrollados.

Tabla III. XXXII: Valoración del Indicador 6.1: Tiempo de Instalación

Tiempo en minutos	Valoración	
	Valoración Cuantitativa	Valoración cualitativa
181 a 240	1	No Satisfactorio

121 a 180	2	Poco Satisfactorio
61 a 120	3	Satisfactorio
10 a 60	4	Muy Satisfactorio

### 3.6.6.1. Valoraciones

Tabla III. XXXIII: Resultados del Parámetro 6: Instalación

Indicadores	Moodle		Dokeos	
	Valor Cualitativo	Valor Cuantitativo	Valor Cualitativo	Valor Cuantitativo
Tiempo de Instalación	Muy Satisfactorio	4	Satisfactorio	3

### 3.6.6.2. Interpretación

**Tiempo de Instalación:** El tiempo de instalación de los componentes depende de si está completamente correcta su estructura y codificación. Los componentes desarrollados en Moodle tuvieron un tiempo de instalación de 4 minutos, con 4 puntos equivalentes a Muy Satisfactorio. Mientras tanto Dokeos por su estructura demora más la instalación de sus componentes demorándose 6 minutos por ello tiene 3 puntos equivalentes a Satisfactorio.

### 3.6.6.3. Calificación

#### Cálculo de Porcentajes

$$Cm = \sum U \qquad Cd = \sum V \qquad Ct = \sum W$$

$$Pm = \left(\frac{Cm}{Ct}\right) * 100\%$$

$$Pd = \left(\frac{Cd}{Ct}\right) * 100\%$$

*Cm*: 4

*Cd*: 3

*Ct*: 4

$$Pm: \left(\frac{4}{4}\right) * 100\% = 100\%$$

$$Pd: \left(\frac{3}{4}\right) * 100\% = 75\%$$

### 3.6.6.4. Representación

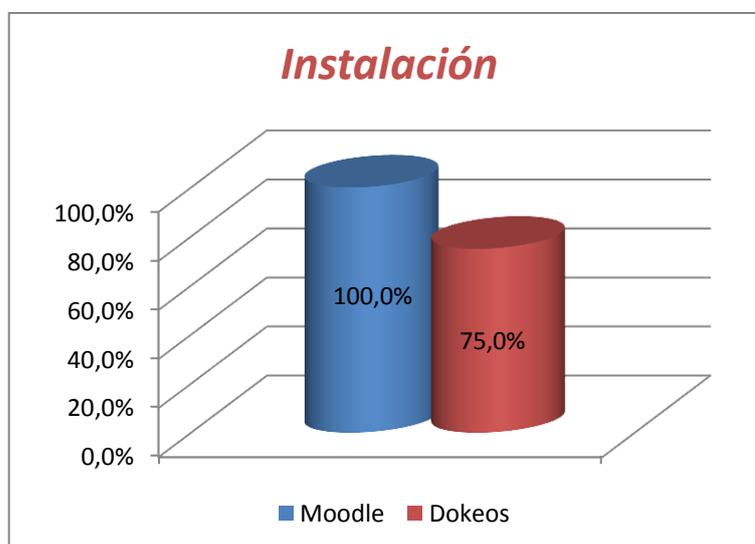


Ilustración III. 43: Representación de Resultados - Parámetro 6: Instalación

### 3.6.7. Puntajes Alcanzados

Tabla III. XXXIV: Cuadro Resultados de Parámetro e Indicador

<i>PARAMETROS</i>	<i>INDICADORES</i>	<i>Moodle</i>	<i>Dokeos</i>
Acceso a Base de Datos	• Soporte para múltiples bases de datos.	4	1
	• Manipulación con la base de datos.	4	3
	• Desempeño con la base de datos.	4	3
Facilidad de Mantenimiento	• Facilidad de manejo de la estructura del LMS (Accesibilidad).	20	17

<i>PARAMETROS</i>	<i>INDICADORES</i>	<i>Moodle</i>	<i>Dokeos</i>
	• Escalabilidad (Actualizaciones).	17	15
	• Tiempo requerido en comprender el código (código legible)	17	15
	• Especificaciones (versiones-compatibilidad) y estándares	20	17
Codificación	• Adaptación de hojas de estilo	19	15
	• Estructura de directorios de los componentes del LMS	19	15
	• Errores, fiabilidad (ausencia de fallos)	17	15
	• Funcionalidad	19	15
	• Tiempo de desarrollo	3	2
Reutilización	• Herencia de clase base	19	13
	• Reutilización de código	19	19
	• Tamaño de la aplicación (componentes)	19	17
Seguridad	• Utilización de roles y capacidades	19	19
	• Integridad de datos	19	19
Instalación	• Tiempo de instalación	4	3
<b>Totales</b>		<b>262</b>	<b>223</b>

### 3.6.8. Diagrama General de Resultados

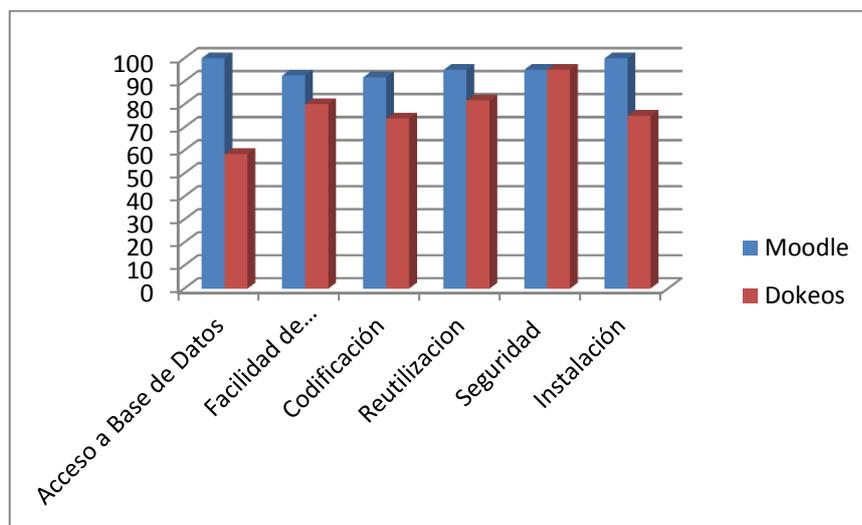


Ilustración III. 44: Gráfica General de Resultados

Tabla III. XXXV: Resultados Generales

	Acceso a Base de Datos	Facilidad de Mantenimiento	Codificación	Reutilización	Seguridad	Instalación
Moodle	100	92,5	91,66	95	95	100
Dokeos	58,33	80	73,80	81,66	95	75

### 3.6.9. Interpretación

Con la tabulación de los resultados obtenidos en la evaluación de los LMS, se da a conocer de una manera global el comportamiento de cada herramienta respecto a la productividad en agregación de componentes.

El Parámetro Acceso a Base Datos fue el encargado de probar la versatilidad con que cada herramienta se comporta al momento de actuar con los datos y los motores que

gestionan estos datos, en donde se destaca Moodle mostrando un mejor comportamiento y ágil creación de los modelos.

Facilidad de Mantenimiento fue el parámetro que pudo darnos una mejor perspectiva de la capacidad y facilidad de mantenimiento en cuanto a componentes existentes, mostrando Moodle una ventaja ante Dokeos.

Codificación es el parámetro que nos permite enfocarnos en la fácil y rápida creación de componentes en cada herramienta, las ventajas que nos brinda determinado LMS en su codificación en donde Moodle presenta virtudes que hacen que sobresalga ante Dokeos.

La reutilización es un parámetro muy importante el mismo que es muy determinante en la agregación de componentes, mediante la valoración de los indicadores en la reutilización pudimos observar que Moodle es mejor en cuanto la reutilización tanto en la utilización de funciones como de código y clases base.

La seguridad está definida por la capacidad de las herramientas en controlar roles y capacidades, y ya que ambas herramientas LMS presentan características similares en el manejo de los indicadores de seguridad, tanto como Moodle así como Dokeos permiten aprovechar las ventajas de un correcto sistema de autenticación y seguridad en el desarrollo y agregación de componentes.

Un proceso de instalación define la versatilidad de un componente y describe un mejor concepto en la utilización e implementación del mismo. Moodle presenta una mejor adaptación en el desarrollo de componentes y más óptimo proceso de instalación destacándose sobre Dokeos.

### **3.7. Análisis de Resultados**

Los resultados que presenta cada LMS al establecer mismos criterios y escenarios de desarrollo son claros ya que ambas herramientas presentan características similares se

puede establecer el LMS que se adapta de mejor manera a la agregación de nuevos componentes.

Luego de la interpretación de los resultados realizado anteriormente se destaca el LMS Moodle para realizar un desarrollo y agregación de componentes más productivo, ideal para desarrollar un módulo prototipo adaptado al nuevo sistema de evaluación (calificación) estudiantil por parte de los Docentes de la Escuela Superior Politécnica de Chimborazo, con los objetivos planteados al principio de la tesis, ya que Moodle presenta grandes características en el desarrollo y agregación de nuevos componentes, manejo de roles y capacidades y un óptimo proceso de instalación.

En conclusión por tener características que permiten distinguir una ventaja sobre Dokeos, concluimos que el LMS Moodle es el más adecuado para la agregación de componentes o módulos.

### 3.8. Comprobación de Hipótesis

La hipótesis planteada es:

**H1:** La agregación de módulos en Moodle es más Productiva que los demás LMS analizados.

#### 3.8.1. Resultados Finales

Tabla III. XXXVI: Resultados finales

<i>PARAMETROS</i>	<i>INDICADORES</i>	<i>Moodle</i>	<i>Dokeos</i>
Acceso a Base de Datos	• Soporte para múltiples bases de datos.	4	1
	• Manipulación con la base de datos.	4	3
	• Desempeño con la base de datos.	4	3
Facilidad de Mantenimiento	• Facilidad de manejo de la estructura del LMS (Accesibilidad).	20	17

<i>PARAMETROS</i>	<i>INDICADORES</i>	<i>Moodle</i>	<i>Dokeos</i>
	• Escalabilidad (Actualizaciones).	17	15
	• Tiempo requerido en comprender el código (código legible)	17	15
	• Especificaciones (versiones-compatibilidad) y estándares	20	17
Codificación	• Adaptación de hojas de estilo	19	15
	• Estructura de directorios de los componentes del LMS	19	15
	• Errores, fiabilidad (ausencia de fallos)	17	15
	• Funcionalidad	19	15
	• Tiempo de desarrollo	3	2
Reutilización	• Herencia de clase base	19	13
	• Reutilización de código	19	19
	• Tamaño de la aplicación (componentes)	19	17
Seguridad	• Utilización de roles y capacidades	19	19
	• Integridad de datos	19	19
Instalación	• Tiempo de instalación	4	3
<b>Totales</b>		<b>262</b>	<b>223</b>

Tabla III. XXXVII: Valores y Porcentajes Finales

	<b>TOTAL</b>	
	<b>Valor</b>	<b>Porcentaje</b>
<b>Moodle</b>	262	93.57
<b>Dokeos</b>	223	79,64

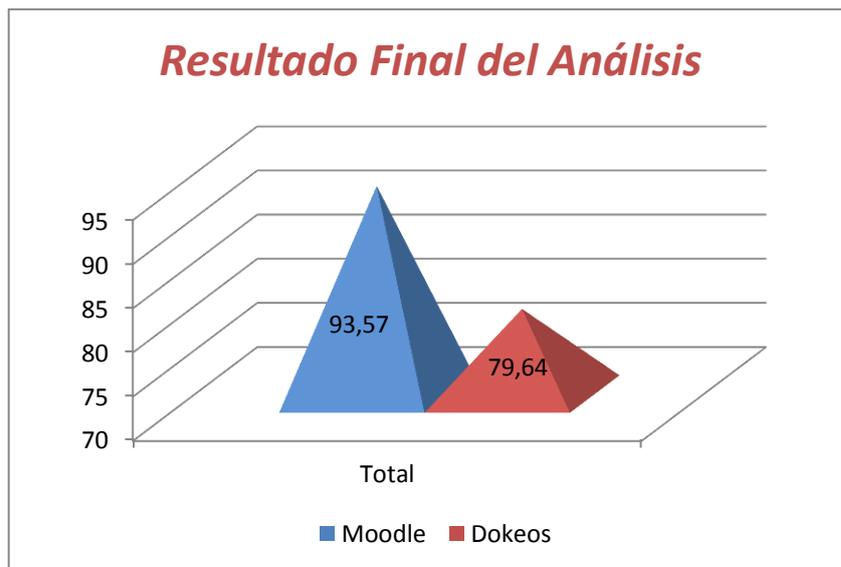


Ilustración III. 45: Resultado Final del Análisis

Haciendo referencia a la Tabla III.XXXVII y por observación directa se concluye que en la hipótesis planteada: Moodle posee el valor más alto 262 puntos sobre 223, valor máximo posible, superando en un 13,93% a la herramienta Dokeos, Moodle tiene el nivel de desempeño del 93,57% que equivale a Muy bueno. Por lo tanto se concluye que la hipótesis H1 es verdadera.

## **CAPÍTULO IV:**

### **DESARROLLO DEL NUEVO MODULO PROTOTIPO ORIENTADO A LA NUEVA GESTIÓN DE CALIFICACIONES EN LA ESPOCH**

En la actualidad en el desarrollo de módulos personalizados orientados a herramientas LMS no se consideran aspectos como robustez, agilidad y confiabilidad que son indispensables al momento de interactuar con la herramienta, acarreado como problema final que el módulo sea desechado por el usuario final, para evitar esto se procede a diseñar un módulo prototipo personalizado que permita orientar en el desarrollo y a la vez que vaya cumpliendo con el alcance y objetivos planteados.

Para resolver el problema del cálculo de las notas de los estudiantes, se desarrolla un nuevo componente prototipo, en conjunto con normas e instructivos para el uso del mismo, y así dar facilidad de cálculos.

#### **4.1. Ingeniería de la Información**

La metodología usada para el desarrollo del componente es SCRUM, una metodología que permite la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

El modelo de desarrollo SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

SCRUM, más que una metodología de desarrollo software, es una forma de auto-gestión del equipo de programadores. El grupo de programadores decide cómo hacer sus tareas y cuánto van a tardar en ello (Sprint). SCRUM ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro

##### **4.1.1. Definición del Ámbito**

Los roles implicados para el desarrollo de este sistema siguiendo SCRUM serán:

##### **Comprometidos:**

- **Propietario del producto: ESPOCH**, responsable de obtener el mayor valor de producto para los docentes y resto de implicados.
- **Equipo de desarrollo:** Erika Villacrés, Geovanny Silva que es el grupo de trabajo que se encarga del desarrollo del producto.
- **SCRUM Manager:** Ing. Gloria Arcos, gestor del equipo, es responsable del funcionamiento y de la productividad del equipo de desarrollo.

## **4.2. Análisis de Requerimientos**

El análisis de requerimientos para el desarrollo de un Módulo Personalizado se ha realizado de manera colaborativa con los docentes de la ESPOCH.

### **4.2.1. Propósito**

El análisis de requerimientos de software se define con el objetivo de especificar las condiciones de un contrato o un acuerdo entre los miembros del grupo de desarrollo y los futuros usuarios del módulo. Detallando las futuras funcionalidades de software así también como las limitaciones del mismo.

### **4.2.2. Definición del problema**

La Escuela Superior Politécnica de Chimborazo, se encuentra en proceso de Acreditación y Calidad Universitaria, por lo cual ha implementado un nuevo sistema de evaluación estudiantil, dividiendo la nota acumulada 28 puntos en tres aportes, primer aporte sobre 8, segundo aporte sobre 10 puntos y tercer aporte sobre 10 puntos; lo cual cambia la temática de evaluación mantenida por algún tiempo de algunos docentes, lo cual les lleva a requerir cambios al entorno virtual.

### **4.2.3. Alternativas de solución**

Con la situación actual en la que se encuentra la Escuela Superior Politécnica de Chimborazo ESPOCH a través de sus Docentes es necesario desarrollar un módulo prototipo adicional orientado a la gestión de calificaciones en el entorno virtual, el mismo que permitirá automatizar el proceso de evaluación virtual y se registrará al nuevo sistema de evaluación estudiantil, el cual contará con una interfaz amigable con el usuario. El módulo facilitará la administración de porcentajes equivalentes en cada evaluación, trabajo, tarea, etc.

El nuevo módulo prototipo a desarrollar permitirá obtener las notas resultantes de los diferentes aportes (foros, tareas, pruebas on-line, etc.) de los alumnos, realizar el cálculo

correspondiente y dar la nota de los diferentes aportes divididos de la siguiente manera: primer aporte sobre 8, segundo aporte sobre 10 puntos y tercer aporte sobre 10 puntos dando un acumulado de 28 puntos.

#### **4.2.4. Descripción General**

Se pretende representar las funciones y características del módulo de forma amplia y detallada, mediante un proceso previo de análisis de requerimientos que permitirá identificar a personas y recursos, así como también analizar la complejidad de los procesos; para de esta manera poder desarrollar un producto de calidad.

#### **4.2.5. Funciones del Módulo**

El módulo ESPOCH proporciona dos funciones que son:

- Cálculo de los parciales en un determinado tiempo.
- Mostrar los resultados en pantalla

### **4.3. Estudio de Factibilidad**

#### **4.3.1. Factibilidad técnica**

##### **a) Recurso Humano**

Ing. Gloria Arcos	Directora de Tesis
Dr. Julio Santillán	Miembro de tesis
Erika Villacres	Desarrollador
Geovanny Silva	Desarrollador

##### **b) Recursos de hardware**

Computadora portátil Hp pavilion g4 Intel Core I5 de 4GB

Memoria RAM de 4GB

500 GB en disco duro.

**c) Recurso de software**

Sistema Operativo: Microsoft Windows 7 / Ultimate x64 bit

Herramienta para desarrollo: Moodle 2.3

PHP 5.4.3

Servidor Web: Apache 2.2.22

MySQL 5.5.24

**4.3.2. Factibilidad operativa**

El módulo que se desarrollará, está destinado a ser operado por cualquier persona que desee tener un módulo personalizado para el cálculo de sus notas, haciendo de forma más fácil la administración de evaluaciones.

**4.4. Desarrollo del Módulo**

**4.4.1. Visión y Planificación**

**Estructura del proyecto**

Esta sección emboza la organización del equipo para el proyecto de desarrollo del Módulo, incluyendo un alto detalle de la visión y alcance del proyecto.

**Visión**

Contar con un módulo personalizado que aproveche los beneficios de la herramienta Moodle, permitiendo contar con un software que optimice los procesos de cálculo de notas.

## Alcance del proyecto

Se pretende contar con un módulo personalizado el cual se capaz de realizar cálculos de notas de forma fácil y segura en periodos de tiempos establecidos, para posteriormente mostrarlos en pantalla.

**Nombre del proyecto:** ESPOCH

## Conjunto de características de alto nivel

- Se debe aprovechar las bondades que pone a su disposición la herramienta Moodle para la agregación de módulos.
- En lo posible el componente debe poseer una estructura modular.

Tabla IV. I: iteraciones

Iteraciones	Tiempos de Duración	Áreas
Iteración 0	40 días	Investigación
Iteración 1	15 días	Php en Moodle
Iteración 2	7 días	Aplicación

## 4.4.2. Análisis y Diseño

### 4.4.2.1. Definición de Usuarios del Modulo

Cada una de las personas determinadas cumple un rol diferente dentro del módulo las cuales son descritas de acuerdo a su nivel de permiso.

**Administrador:** encargado de:

- Gestionar los recursos de la plataforma
- Establecer fechas topes para la emisión de los parciales.

**Docente:** encargado de:

- Establecer lecciones, foros, pruebas, etc. Dentro del entorno virtual
- Evaluar o establecer parámetros de evaluación.

#### 4.4.2.2. Análisis del sistema

Para una mejor comprensión de los requerimientos planteados se los representara en los principales diagramas necesarios para la ingeniería de software descripciones narrativas y/o graficas de cada proceso.

### CASO DE USO CÁLCULO PARCIALES

Tabla IV. II: Caso de Uso Cálculo Parciales

<b>IDENTIFICAR CASO DE USO:</b>	CU_CALCULO_PARCIALES
<b>NOMBRE DEL CASO DE USO:</b>	Cálculo Parciales
<b>ACTORES:</b>	Administrador, Docente
<b>PROPÓSITO:</b>	Realizar cálculo y Mostrar en pantalla
<b>VISIÓN GENERAL:</b>	El proceso inicia cuando el docente utiliza el módulo internamente se ejecuta el cálculo y muestra en pantalla
<b>TIPO:</b>	Primario, real y expandido
<b>REFERENCIAS:</b>	Requerimientos
<b>CURSO TÍPICO DE EVENTOS</b>	
<b>ACCIÓN DEL ACTOR</b>	<b>RESPUESTAS DEL SISTEMA</b>
1.- Se inicia cuando se requiere obtener los parciales	
2.- Usuario usa el módulo	
3.- Se realiza el cálculo	4.-Solicita fechas de cada periodo
5.- Muestra en pantalla	
<b>CURSOS ALTERNATIVOS</b>	

#### Definir los Casos de Uso

Esta actividad nos permite representar los casos de uso a partir de la nomenclatura definida en UML, tanto para la representación de actores, casos de uso, interacciones y relaciones.

## DIAGRAMA DE CASO DE USO UTILIZACIÓN MÓDULO

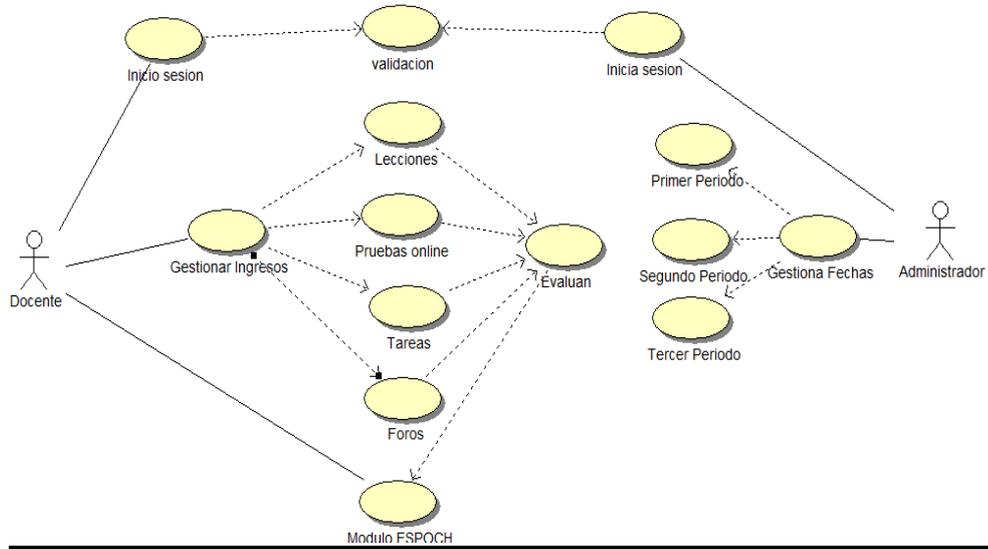


Ilustración IV. 1: Diagrama de Caso de Uso Utilización del Módulo

## Diagrama de Secuencias

## DIAGRAMA DE SECUENCIA UTILIZACIÓN MÓDULO

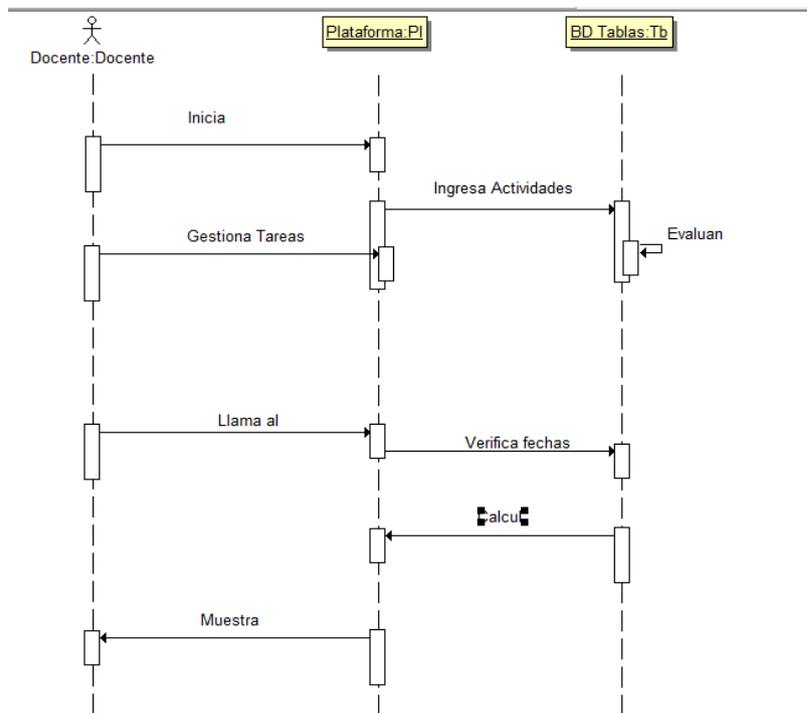


Ilustración IV. 2: Diagrama Secuencia Utilización Módulo

### **4.4.3. Ciclos de Desarrollo – SCRUM**

#### **4.4.3.1. Iteración 0 – Investigación**

##### **Casos de Usuario**

**Se desea investigar la estructura interna de la herramienta LMS Moodle:**

- Directorios
- Archivos de configuración
- Arquitectura
- Requisitos

#### **4.4.3.2. Iteración 1 – Php en Moodle**

Ver anexo 9

#### **4.4.3.3. Iteración 2 – Módulo**

##### **Casos de Usuario**

**Se desea gestionar una pantalla de resultados:**

- Diseño de un reporte
- Lógica de programación en el cálculo
- Adaptación del módulo a la versión

### **4.5. Pruebas**

#### **4.5.1. Pruebas Unitarias**

Para asegurar el correcto funcionamiento del Módulo se han probado sus cálculos de forma independiente, enviando datos de entrada, para luego obtener los a través de los

diferentes métodos para realizar consultas. Se ha probado que el cálculo realizado sea correcto.

#### **4.5.2. Pruebas del Módulo**

Las pruebas finales consistieron en verificar que la información ingresada se vea inmediatamente reflejada en la consulta de modulo.

Se provocaron errores intencionales para verificar el correcto funcionamiento del módulo, así como de las funciones de validación de datos, como por ejemplo:

- Realizar consultas a evaluaciones vacías
- Ingreso de notas fuera de los periodos

## CONCLUSIONES

1. El análisis de las Herramientas LMS Moodle y Dokeos nos permite obtener una clara visión en el enfoque de una agregación más productiva de nuevos módulos o componentes.
2. Tanto Moodle así como Dokeos presentan similares características las cuales nos brindan un óptimo beneficio al momento de agregar nuevos componentes, diferenciándose entre sí más no por su conceptualidad sino por su implementación y estructura.
3. El realizar un análisis de la productividad de agregación de componentes permite determinar con claridad las prestaciones que nos brinda un componente como producto final, y debido a su importancia se ha considerado los factores más importantes entre los cuales destacamos: Acceso a Base de Datos, Facilidad de Mantenimiento, Reutilización, Seguridad y Codificación.
4. Al realizar el estudio en los 2 LMS se puede determinar que de acuerdo al desarrollo de los prototipos de prueba agregados, Dokeos presenta 58% contra Moodle con el 100% demostrando mejores características respecto al acceso de los datos.
5. El estudio de la Facilidad de Mantenimiento es muy importante puesto que refleja la escalabilidad de los componentes. Al ser agregados los prototipos de prueba Moodle presenta un 92.5% frente al 80% que presenta Dokeos, demostrando Moodle una ventaja de 12.5% que permite determinar su ventaja al momento del mantenimiento.
6. El Codificación es un indicador que determina la funcionalidad de los componentes. Por su mejor adaptación entre el lenguaje y el acceso a datos Moodle muestra un 91,6% de eficiencia frente a Dokeos con un 73,8% con una ventaja de 17.8%.

7. El indicador de Reutilización demuestra la correcta utilización de objetos (clases y métodos) que pueden ser reutilizados lo cual permite aprovechar el trabajo anterior, economizar tiempo, y reducir la redundancia. Para lo cual Moodle con 95% de eficiencia mostrando una ventaja de 13,4% frente a 81,6% de Dokeos.
8. El estudio de la Seguridad de los componentes o módulos demanda una correcta utilización de factores como Roles y Capacidades que determinan un correcto cuidado de la información que se mostrará. Ambas herramientas LMS presentan características similares las que representan un 95% de eficiencia en la seguridad de módulos.
9. La investigación realizada acerca de la productividad en la agregación de componentes o módulos permite determinar la herramienta apropiada para el desarrollo del nuevo módulo prototipo ESPOCH el mismo que de acuerdo al estudio efectuado se lo ha desarrollado con herramienta Moodle bajo PHP con MySQL. Este módulo permitirá mostrar a los Docentes las calificaciones de los estudiantes según el nuevo sistema de calificación vigente en la Escuela Superior Politécnica de Chimborazo.

## RECOMENDACIONES

1. Al realizar un análisis comparativo de productividad en la agregación de componentes, se debe escoger muy bien los parámetros de comparación, se recomienda que entre los parámetros se considere: la codificación y la seguridad, también se pueden establecer parámetros que permitan probar el acoplamiento existente con plantillas HTML que usen estilos, la reutilización de código, etc., en fin se deben comparar las características necesarias para llegar a una agregación de componentes productiva.
2. Es recomendable destacar la importancia de definir escenarios de prueba bajo las mismas características ya que nos permitirán establecer de mejor manera y con hechos reales, las características y beneficios que nos proporciona cada LMS para la agregación de componentes.
3. Para explotar de mejor manera los beneficios de utilizar estas tecnologías lo recomendable es tener conocimientos de PHP, Hojas de estilo CSS, además de MySQL.
4. Se recomienda utilizar este tipo de LMS para mejorar la productividad en la agregación de componentes ya que ayudan, al brindarnos facilidades como son las de evitarnos programar las conexiones a Base de Datos, Autenticación, métodos de Ingreso, etc.
5. Para proyectos en los que el tiempo de desarrollo es un factor importante se recomienda la utilización de la metodología SCRUM, ya que al combinarla con la utilización de un LMS como Moodle o Dokeos, finalmente obtendremos una aplicación en un tiempo considerablemente menor, además de estar interactuando constantemente durante el desarrollo con los futuros usuarios del componente.
6. Se recomienda también la utilización de herramientas software libre, ya que no estamos limitados a un software en específico, sino que estamos en un ambiente en el que todos podemos colaborar con nuestros aportes, podemos compartir y mejorar

lo que nos han dado y lo más importante sabremos el porqué de las cosas, ya que al tener la cultura de software libre siempre habrá personas dispuestas en colaborar con sus conocimientos hacia la comunidad.

## RESUMEN

Análisis de la Productividad para agregar componentes en las herramientas LMS aplicado al Desarrollo de un nuevo módulo orientado a la nueva gestión de calificaciones de la ESPOCH.

Se aplicó el método descriptivo para evaluar ciertas características mediante la Metodología SCRUM , para el desarrollo del módulo prototipo se utilizó herramientas de software libre Dreamweaver 8.0 para la codificación, Moodle 2.3 con Php5.4.3, MySQL 5.5.24, y Servidor Apache 2.2.22, bajo Sistema Operativo Windows 7 64 bits. Se estableció parámetros de comparación para determinar las dos herramientas más aptas para la agregación de componentes entre Moodle, Dokeos, Claroline, ATutor y Sakai, evaluando su Madurez del Producto y Personalización, determinamos a Moodle y Dokeos.

Se realizó módulos prototipos de similares características en las herramientas LMS (Sistema de gestión de aprendizaje) Moodle y Dokeos.

De acuerdo al análisis comparativo mediante parámetros: Acceso a la base de Datos, Facilidad de Mantenimiento, Codificación, Reutilización de código, Seguridad, Instalación, nos da un 93.57% para la Herramienta LMS Moodle y 79,64% para la Herramienta LMS Dokeos, concluyendo que Moodle es el LMS que más prestaciones brinda al momento del desarrollo de los nuevos módulos.

Concluimos de esta manera que Moodle es el LMS (Sistema de gestión de aprendizaje) más productivo en la agregación de nuevos módulos o componentes.

Se recomienda el uso del componente prototipo orientado a la nueva gestión de calificaciones de la ESPOCH, que facilitará al Docente el cálculo de las notas de los estudiantes.

## **ABSTRACT**

Analysis of Productivity to add components in the LMS tools applied to the Development of a new module oriented to a new grades management of ESPOCH.

It was applied the descriptive method to evaluate several characteristics by using the SCRUM Methodology, for the development of de prototype module it was used free software tools such as: Dreamweaver 8.0 for the codification , Moodle 2.3 with Php 5.4.3, MySQL 5.5.24, and Apache Server 2.2.22, under the Windows 7 Operative System of 64 bits.

It was established comparison parameters to determine the two tools more apt to add the components among Moodle, Dokeos, Claroline, ATutor and Sakai, evaluating its Product Maturity and Personalization; it is determined Moodle and Dokeos. It was made prototype modules of similar characteristics within the LMS tools (Learning Management System) Moodle and Dokeos. According to the comparative analysis through parameters: Access of the Databases, Ease of Maintenance, Codification, Code Reuse, Security, Installing, it gave as result of 93,57% for the LMS Moodle Tool and 79,64% for the LMS Tool Dokeos, concluding that Moodle is the LMS that brought higher performance at the moment of developing new module.

It is concluded that Moodle is the LMS ((Learning Management System) more productive in the adding of the new modules or components.

## BIBLIOGRAFÍA

- [1] ACERCA DE MOODLE. Moodle Documentación. Retrieved Marzo 21 de 2012 [en línea], from: <http://docs.moodle.org/es> [Consulta: 21 Marzo 2013]
- [2] ATUTOR. Sitio Web Oficial. [en línea], from: <http://www.atutor.ca> [Consulta: 13 Julio de 2012]
- [3] ARQUITECTURA DE MOODLE. Moodle Documentación. [en línea], from: [http://docs.moodle.org/all/es/Arquitectura\\_de\\_Moodle](http://docs.moodle.org/all/es/Arquitectura_de_Moodle) [Consulta: 23 Marzo 2013]
- [4] BENEFICIOS LMS. 27 noviembre 2012. [en línea], from: <http://webanywhere-espanol.blogspot.com/2012/11/beneficios-de-los-lms.html> [Consulta: 14 Febrero 2013]
- [5] BENEFICIOS PARA EL ESTUDIANTE 2012. [en línea], from: <http://www.cartero.com/servicios/soluciones-web/soluciones-de-e-learning> [Consulta: 14 Febrero 2013]
- [6] BONEU, JOSEP M. (2007). Plataformas abiertas de e-learning para el soporte de contenidos educativos abierto. Revista de Universidad y Sociedad del Conocimiento. [monográfico en línea], from: <http://www.uoc.edu/rusc/4/1/dt/esp/boneu.pdf> [Consulta: 8 Junio 2012]
- [7] BOWLES MARCUS, 2005 Learning to E-Learn Project: Rediscovering the Benefits of E-Learning. [en línea], from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.5106&rep=rep1&type=pdf> [Consulta: 25 Julio 2012]
- [8] CLARA LÓPEZ GUZMÁN. Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. [en línea], from:

[http://gredos.usal.es/jspui/bitstream/10366/56649/1/DIA\\_Repositoriosobjetos.pdf](http://gredos.usal.es/jspui/bitstream/10366/56649/1/DIA_Repositoriosobjetos.pdf) [Consulta: 6 Agosto 2012]

- [9] CLAROLINE. Sitio Web Oficial. [en línea] <http://www.claroline.net> [Consulta: 13 Julio de 2012]
  
- [10] CRUZ HUAMAN EDGAR ALEXANDER, GUERRA FLORES RORIBIO HELWER. Influencia de un sistema gestor del conocimiento para el logro de las capacidades del Área de matemáticas del 5to grado de educación secundaria de la institución educativa particular Pronoe Galileo. Tesis (Ingeniero de Sistemas e Informática [en línea], from: <http://es.scribd.com/doc/30992797/TESIS-Influencia-de-un-Sistema-Gestor-del-Conocimiento-para-el-logro-de-las-Capacidades-del-area-de-matematicas-del-5to-grado-de-Educacion-Secundaria#download> [Consulta: 2 Mayo 2012]
  
- [11] DIEGO MACIAS ALVAREZ Plataformas de enseñanza virtual libres y sus características de extensión: Desarrollo de un bloque para la gestión de tutorías en Moodle. Diciembre 2010. [en línea], from: <http://www2.uah.es/libretics/files/Tutorias.pdf> [Consulta: 4 Diciembre 2012]
  
- [12] DOKEOS. Sitio Web Oficial. [en línea] <http://www.dokeos.com> [Consulta: 3 Agosto de 2012]
  
- [13] eLearning WORKSHOPS. Aprendizaje virtual, Beneficios de la tecnología en el campo educativo. [en línea], from: <http://www.elearningworkshops.com/modules.php?name=News&file=article&sid=239> [Consulta: 4 Agosto 2012]
  
- [14] ESTADÍSTICAS DE MOODLE. [en línea], [www.http://moodle.org/stats/](http://moodle.org/stats/) [Consulta: 14 noviembre 2012]

- [15] ESTILO DE CODIGO MOODLE: Moodle Documentación. en línea], from: [http://docs.moodle.org/all/es/Manual de Estilo de Código](http://docs.moodle.org/all/es/Manual_de_Estilo_de_Código) [Consulta: 23 Marzo 2013]
- [16] EDUMILIS M. MÉNDEZ, MARÍA A. PÉREZ, ANNA C. GRIMÁN, LUIS E. MENDOZA. ¿Cómo se relacionan la Calidad Sistémica y la Productividad en el Proceso de Desarrollo de Software? Caracas Venezuela, Universidad Simón Bolívar. [en línea], from: [http://www.iiisci.org/journal/CV\\$/risi/pdfs/P799708.pdf](http://www.iiisci.org/journal/CV$/risi/pdfs/P799708.pdf) [Consulta: 23 Marzo 2013]
- [17] JORGE DAVID CASTILLO ESPEJO. L.M.S. Y WEB 2.0 Aprender y colaborar por Internet C.U.N. 2011. [en línea], from: <http://lmsyweb20.wikispaces.com/file/view/LMS+Y+WEB+2.0.pdf> [Consulta: 26 Junio 2012]
- [18] LUIS ALFREEDO GONZALES GOMEZ. Universidad a distancia para el aprendizaje en línea, Ingeniería On-Line. 2006. [en línea], from: [http://biblioteca.usac.edu.gt/tesis/08/08\\_0333\\_CS.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0333_CS.pdf) [Consulta: 15 octubre 2012]
- [19] MANUAL DESARROLLADOR DOKEOS. Sitio Web oficial. [en línea], from: <http://www.dokeos.com/doc/DokeosDevManSpanish15.pdf> [Consulta: 23 Marzo 2013]
- [20] MARC J. ROSENBERG. “Beyond E-Learning: Approaches and Technologies to Enhance Organizational Knowledge, Learning, and Performance”. Editorial Pfeiffer. 2006. [en línea], from: <http://books.google.com.ec/books?hl=es&lr=&id=M5REo6Qj74gC&oi=fnd&pg=PR17&dq=Rosenberg+M.+%E2%80%9CBeyond+E-Learning:+Approaches+and+Technologies+to+Enhance+Organizational+Knowledge,+Learning,+and+Performance&ots=G5phOpKZ2H&sig=BRpYBIPGxXF>

[Rp3hY9xdc0dkCdE#v=onepage&q=Rosenberg%20M.%20%E2%80%9CBeyond%20E-Learning%3A%20Approaches%20and%20Technologies%20to%20Enhance%20Organizational%20Knowledge%2C%20Learning%2C%20and%20Performance&f=false](#) [Consulta: 25 Julio 2012]

- [21] MARÍA JESÚS GARCÍA ALBA. Análisis del desarrollo de extensiones para Moodle: Desarrollo de un módulo para la gestión de laboratorios docentes Diciembre 2010. Proyecto Fin de Carrera (Ingeniería Informática) [en línea], from: <http://www2.uah.es/libretics/files/GruposLab.pdf> [Consulta: 15 octubre 2012]
- [22] MAYURI SOMAYABERI OMAÑA MONTOYA. Modelo de calidad basado en características para la Selección de un Sistema de Gestión de Aprendizaje (LMS- Learning Management System). Trabajo de Grado de Maestría (Postgrado En Sistemas de Información). [en línea], from: <http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAR7606.pdf> [Consulta: 5 Mayo 2012]
- [23] MOODLE. Sitio Web Oficial. [en línea], from: <http://www.moodle.org> [Consulta: 14 Agosto 2012]
- [24] OSI. Open Source Definition, v1.9. [en línea], from: <http://www.opensource.org/docs/osd> [Consulta: 14 noviembre 2012]
- [25] OSI. Open Source Licenses. [en línea], from: <http://www.opensource.org/licenses/index.html> [Consulta: 14 noviembre 2012].
- [27] SAKAI Sitio Web Oficial. [en línea], <http://www.sakaiproject.org> [Consulta: 13 Julio de 2012]
- [28] SISBIB Sistema de Bibliotecas AULAS VIRTUALES COMO HERRAMIENTA DE APOYO EN LA EDUCACIÓN DE LA UNIVERSIDAD

NACIONAL MAYOR DE SAN MARCOS. [en línea], from: [http://sisbib.unmsm.edu.pe/bibvirtual/tesis/Ingenie/Caba%F1as\\_V\\_J/cap1.htm](http://sisbib.unmsm.edu.pe/bibvirtual/tesis/Ingenie/Caba%F1as_V_J/cap1.htm) [Consulta: 4 Agosto 2012]

- [29] THORNE, KAYE. (2003). Blended Learning: How to Integrate Online & Traditional Learning. London: VA Kogan. [en línea], from: <http://kenanaonline.com/files/0011/11429/Blended-Learning.pdf> [Consulta: 6 Agosto 2012].
- [30] WIKIPEDIA Ambiente Virtual Educativo. [en línea], from: [http://es.wikipedia.org/wiki/Ambiente\\_Educativo\\_Virtual](http://es.wikipedia.org/wiki/Ambiente_Educativo_Virtual) [Consulta: 15 Marzo 2012]
- [31] WIKIPEDIA. ATutor. Retrieved Julio 30 de 2012 [en línea], from: <http://es.wikipedia.org/wiki/ATutor> [Consulta: 30 Julio de 2012]
- [32] WIKIPEDIA. Claroline. [en línea], from: <http://es.wikipedia.org/wiki/Claroline> [Consulta: 30 Julio de 2012].
- [33] WIKIPEDIA. Dokeos. [en línea], from: <http://es.wikipedia.org/wiki/Dokeos> [Consulta: 21 Marzo 2013].
- [34] WIKIPEDIA. Software Libre. [en línea], from: [http://es.wikipedia.org/wiki/Software\\_libre](http://es.wikipedia.org/wiki/Software_libre) [Consulta: 14 noviembre 2012].
- [35] WIKIPEDIA. Sakai. [en línea], from: <http://es.wikipedia.org/wiki/Sakai> [Consulta: 13 Julio 2013].

# ANEXOS

## **ANEXO 1:**

### **Instalación de Moodle**

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

Se explicará en detalle el proceso de instalación tanto de Moodle 2.3 como de sus requisitos previos en un servidor Windows, en concreto en Windows 7.

#### **Requisitos:**

Los requerimientos de instalación de Moodle son los siguientes:

- 1. Un servidor web:** La mayoría de los usuarios utilizan Apache, pero Moodle debe funcionar bien en cualquier servidor web que soporte PHP, como el IIS (Internet Information Server) de las plataformas Windows.
- 2. Una instalación de PHP en funcionamiento:** PHP 5 que está soportado a partir de Moodle 1.4. y nuestra versión de Moodle es 2.3.1.
- 3. Una base de datos:** MySQL o PostgreSQL, que están completamente soportadas y recomendadas para su uso con Moodle. MySQL es la elección preferida para mucha gente.

El apartado siguiente es una guía de cómo instalar Apache2, MySQL y PHP5. Proporcionará instrucciones, paso a paso para instalarlos.

Para instalar los requisitos antes mencionados procederemos a descargarnos el paquete WAMP Server que contiene Apache 2 2.22, PHP 5.4.3 y MySQL 5.5.24 de su página oficial [www.wamp.com](http://www.wamp.com), de acuerdo a las características de la maquina en la que se trabajará, en nuestro caso WAMP Server 2.2 para 64 bits, primero nos solicitara descargarnos un paquete adicional como requerimiento también de acuerdo a las características de la maquina donde se instalara WAMP Server 2.2:

## Instalación de WampServer 2.2

Antes de la instalación de WampServer debemos instalar el paquete que nos obligaba tener instalado. Es cual es vcredist\_x64.exe dándolo doble clic.

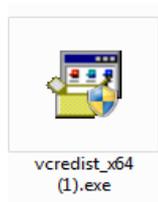


Ilustración I: vcredist\_x64.exe

Procedemos a instalar WAMP Server 2.2, Dando doble clic en el archivo wampserver2.2e-php5.4.3-httpd2.2.22-mysql5.5.24-x64.exe descargado.



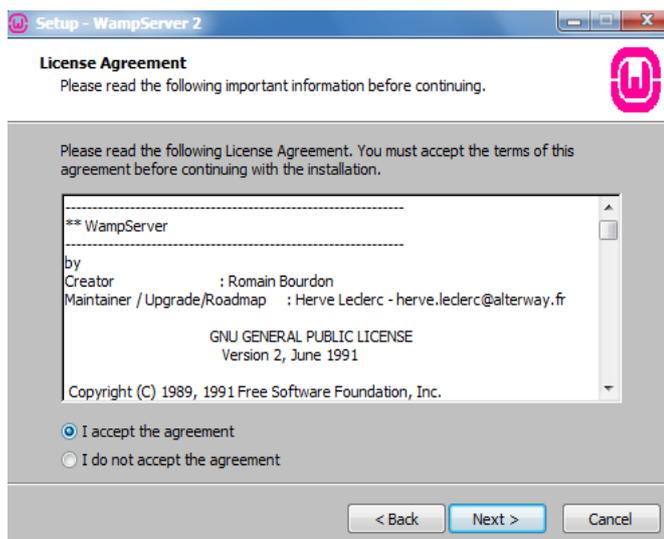
Ilustración II: Archivo de Instalación WAMP Server 2.2

Nos aparecerá la siguiente pantalla, presionamos el botón Next.



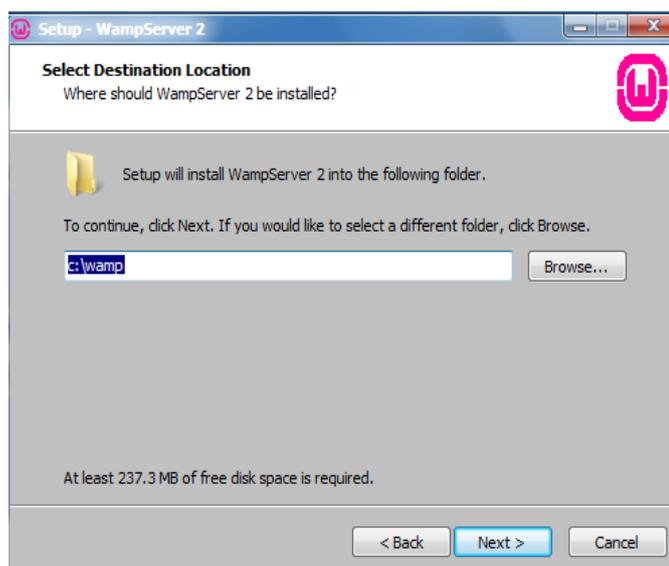
Ilustración III: Instalación WampServer - Paso 1

A continuación nos mostrara la pantalla en a que se debe aceptar la licencia y presionamos el botón Next.



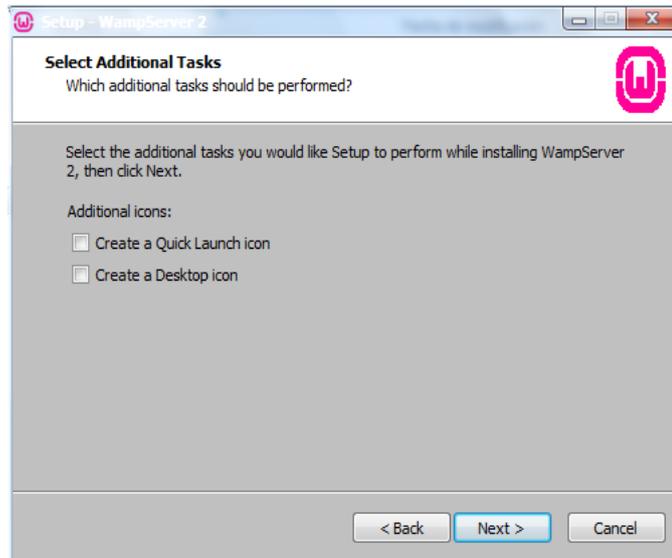
**Ilustración IV: Instalación WampServer – Paso 2**

Posteriormente seleccionamos la ruta en la que deseamos instalar WampServer en nuestro caso en C:\wamp, presionamos la opción Next.



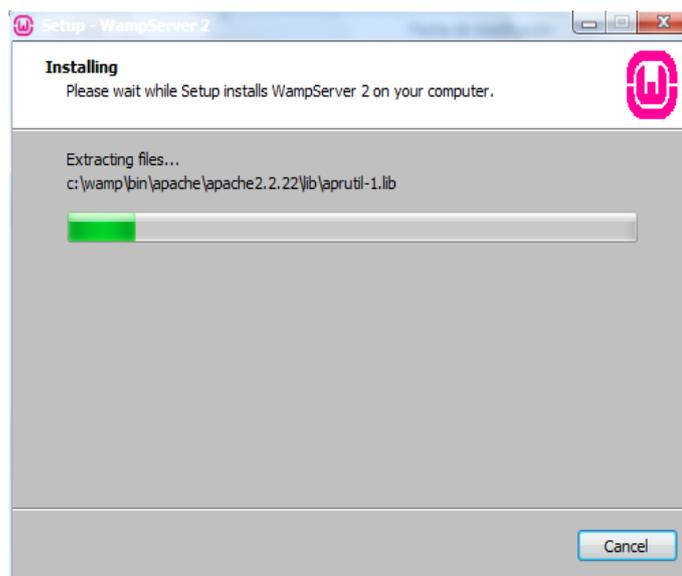
**Ilustración V: Instalación WampServer – Paso 3**

A continuación seleccionamos si deseamos que se cree el acceso directo en el escritorio y si deseamos que se cree el icono en el menú inicio. Presionamos la opción Next.



**Ilustración VI: Instalación WampServer – Paso 4**

Se procede a instalar WampServer.



**Ilustración VII: Instalación WampServer – Paso 5**

Especificamos el Servidor SMTP y el email que será utilizado por PHP si utiliza la función mail(). Establecemos los parámetros por defecto y presionamos la opción Next.

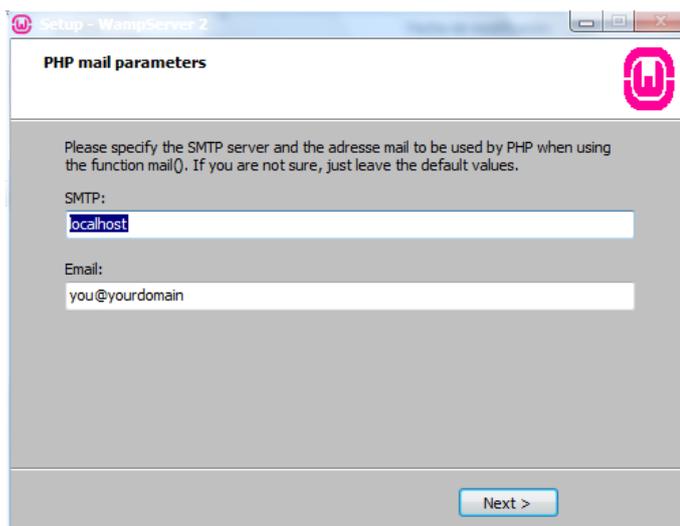


Ilustración VIII: Instalación WampServer – Paso 6

Tenemos la instalado WampServer 2.2, lo iniciamos dando doble clic en el icono en el menú Inicio y tenemos la consola de configuración. Iniciamos los Servicios y encendemos WampServer.

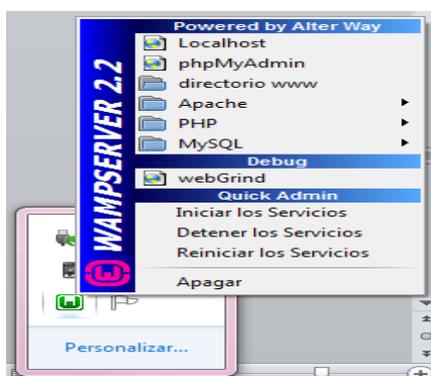


Ilustración IX: Consola de configuración de WampServer

Cambiamos el idioma dando clic derecho en el icono de WampServer, en la opción idioma en nuestro caso Spanish.

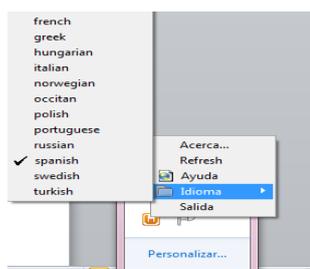


Ilustración X: Configuración de idioma de WampServer

Ingresamos al navegador, pondremos la IP del servidor en la barra de direcciones: localhost. Y observamos a WampServer instalado correctamente. Y procedemos a instalar Moodle.

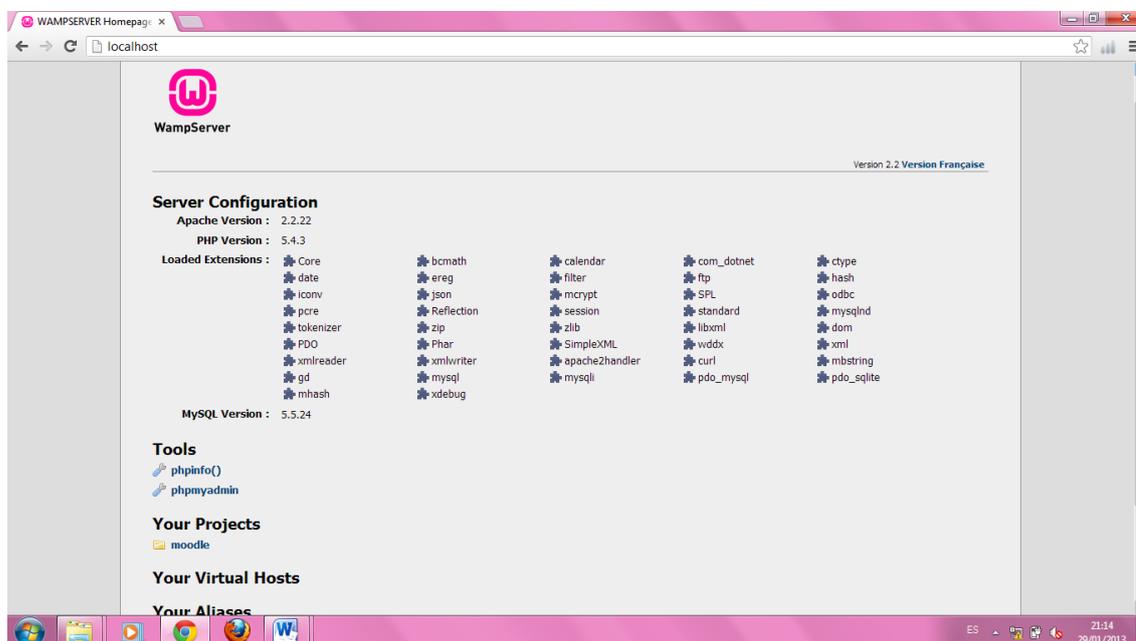


Ilustración XI: WampServer instalado correctamente

## Instalación de Moodle

Después de la instalación en nuestro servidor Wamp, podemos proceder a instalar nuestra plataforma Moodle. Para ello debemos descargar la última versión estable (en nuestro caso Moodle v 2.3) de la página oficial de descargas de Moodle:

A continuación descomprimos el fichero y movemos la carpeta moodle a C:/wamp/www/ que es donde, por defecto, se alojan las páginas web en Apache2:

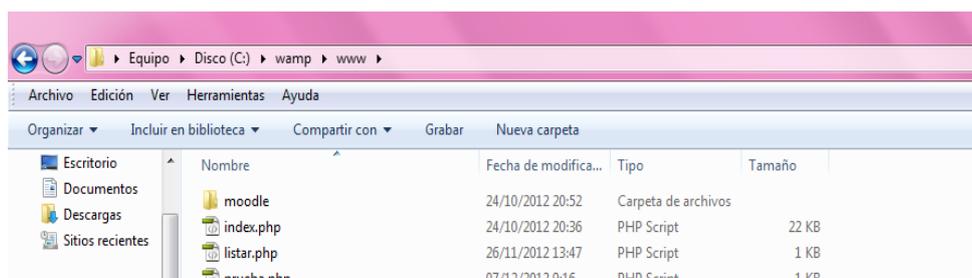


Ilustración XII: Descomprimir Moodle y colocarlo el directorio www

Una vez ejecutado paso anterior ya dejamos de lado el Terminal y los siguientes pasos los realizaremos desde un navegador web, donde pondremos la IP del servidor en la barra de direcciones: localhost/Moodle nos expondrá la siguiente pantalla



Ilustración XIII: Instalación de Moodle-Paso 1

Después de seleccionar el idioma, hacemos clic en Siguiete para continuar la instalación. A continuación el sistema comprueba si tenemos todo lo necesario para que Moodle se ejecute correctamente. Y confirma las rutas, donde se instalará Moodle, la ruta del directorio de datos será C:\wamp/moodledata.

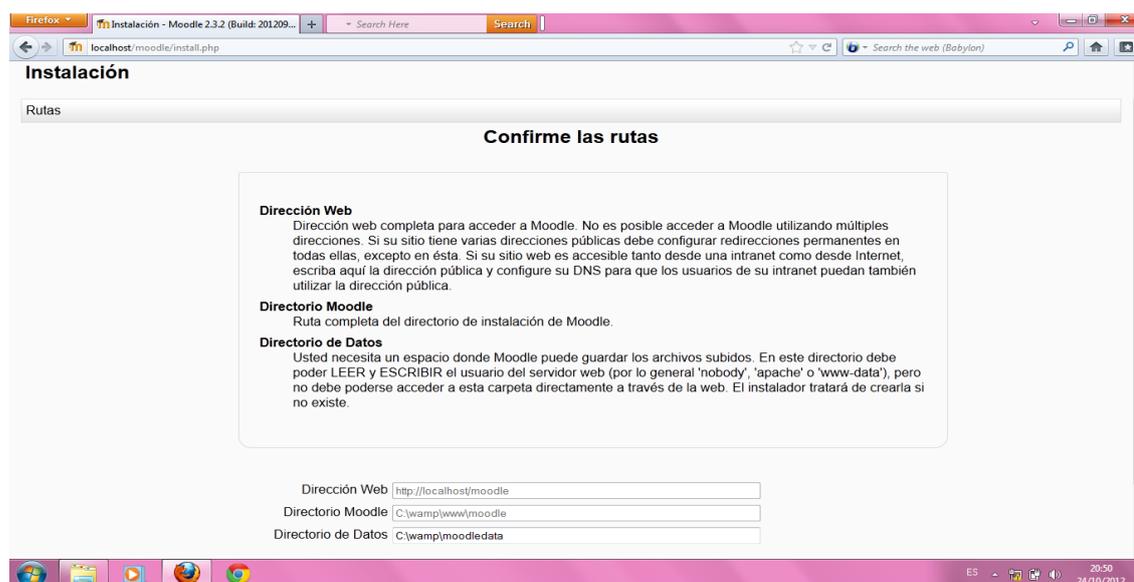


Ilustración XIV: Instalación de Moodle - Paso 2

Después procedemos a seleccionar el controlador de base de datos, seleccionamos la opción Improved MySQL (native/mysql) y presionamos la opción siguiente.

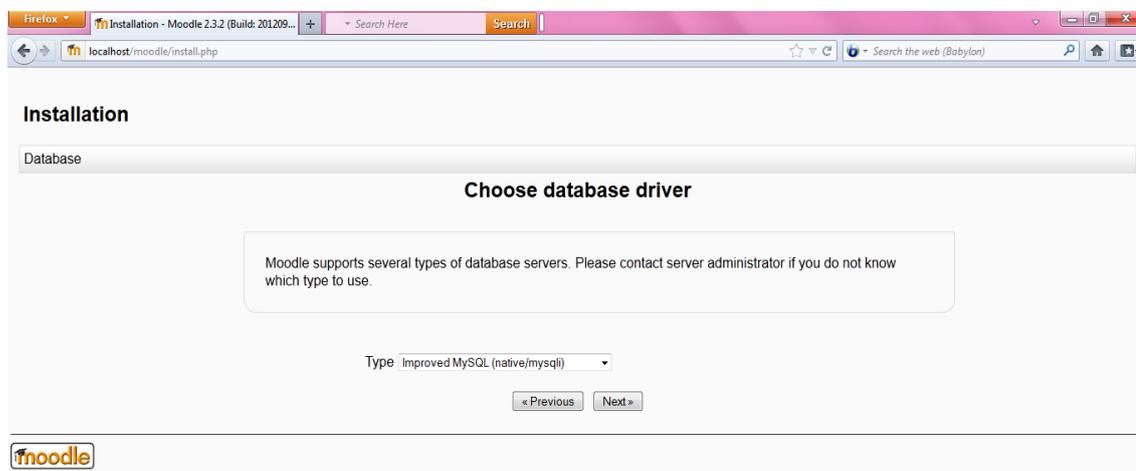


Ilustración XV: Instalación Moodle - Paso 3

Tenemos que completar la información correspondiente a MySQL como el nombre de usuario, Nombre de la base de datos y el prefijo de las tablas de la base de datos.

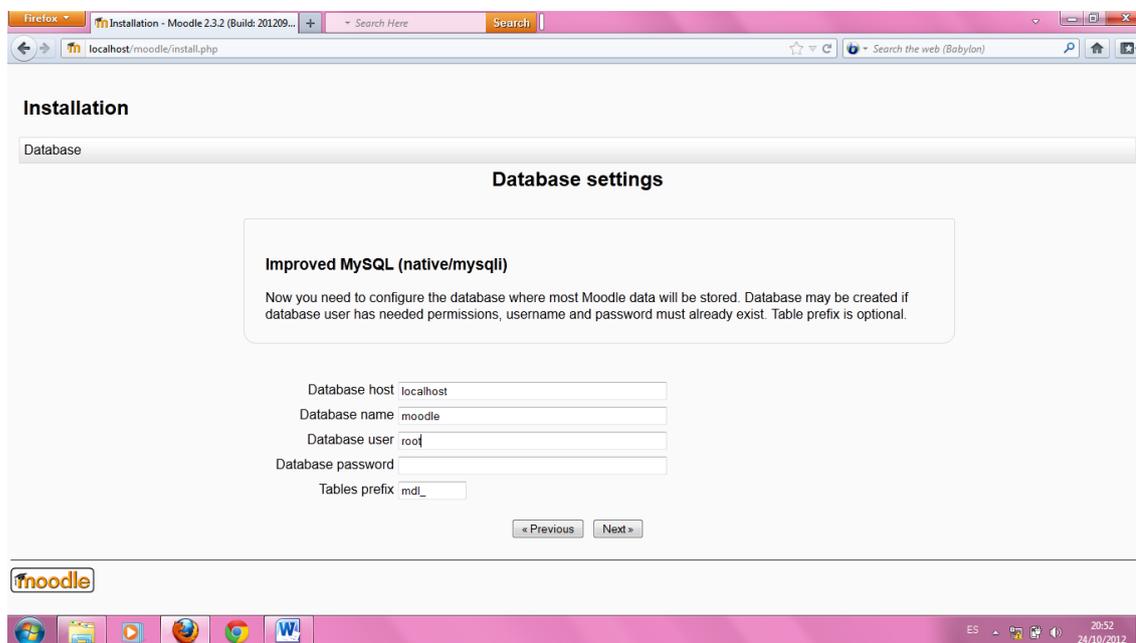


Ilustración XVI: Instalación de Moodle - Paso 4

Aceptamos las condiciones de la licencia y presionamos la opción continuar.

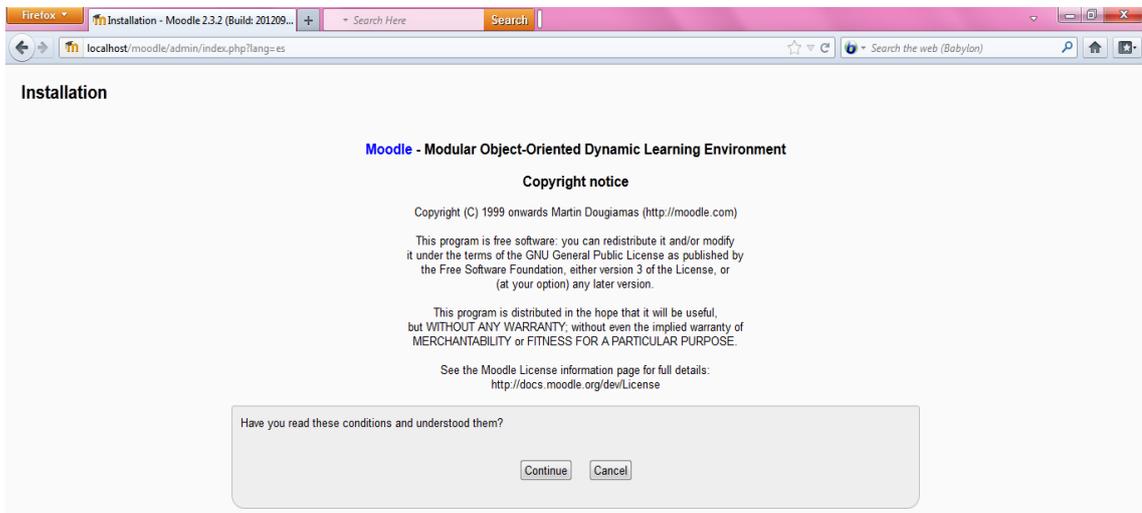


Ilustración XVII: Instalación de Moodle - Paso 5

A continuación se verifica que se cumplen todos los requisitos necesarios en la configuración del servidor. Si no es así, podemos abrir un terminal e instalar los paquetes requeridos o recomendados antes de continuar con la instalación de Moodle.

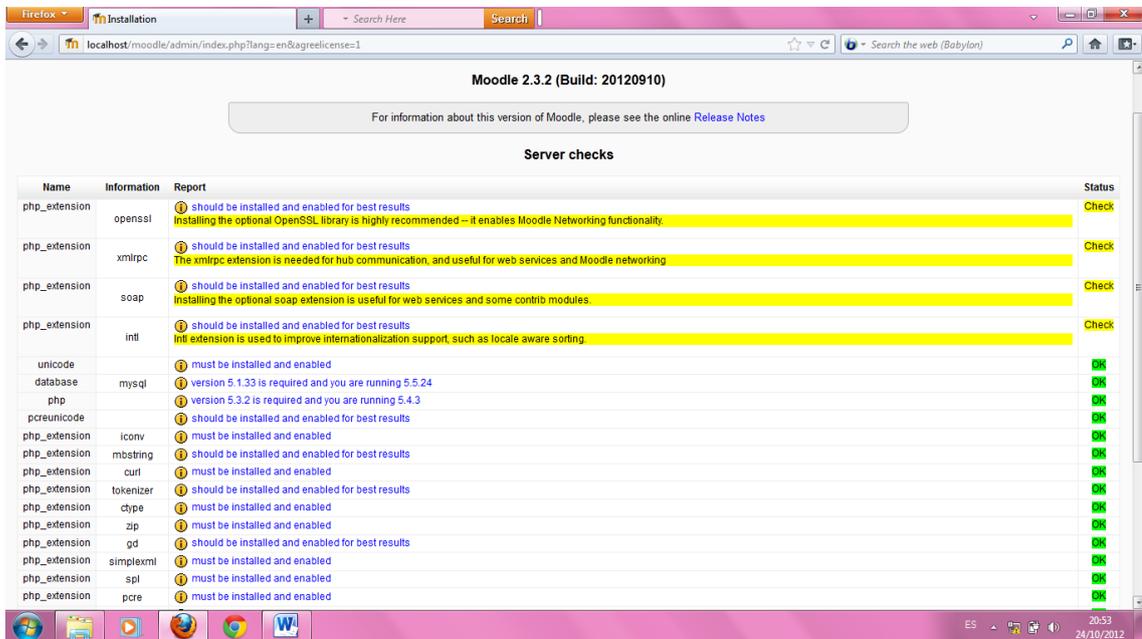


Ilustración XVIII: Instalación de Moodle - Paso 6

En el paso siguiente descargamos el paquete de idioma de español Internacional (es.zip). Durante todo el proceso siguiente aparecerán pantallas que nos informarán de la inserción de nuevas tablas en la base de datos. Debemos comprobar que no existan mensajes de error y aceptar para pasar a la siguiente pantalla.

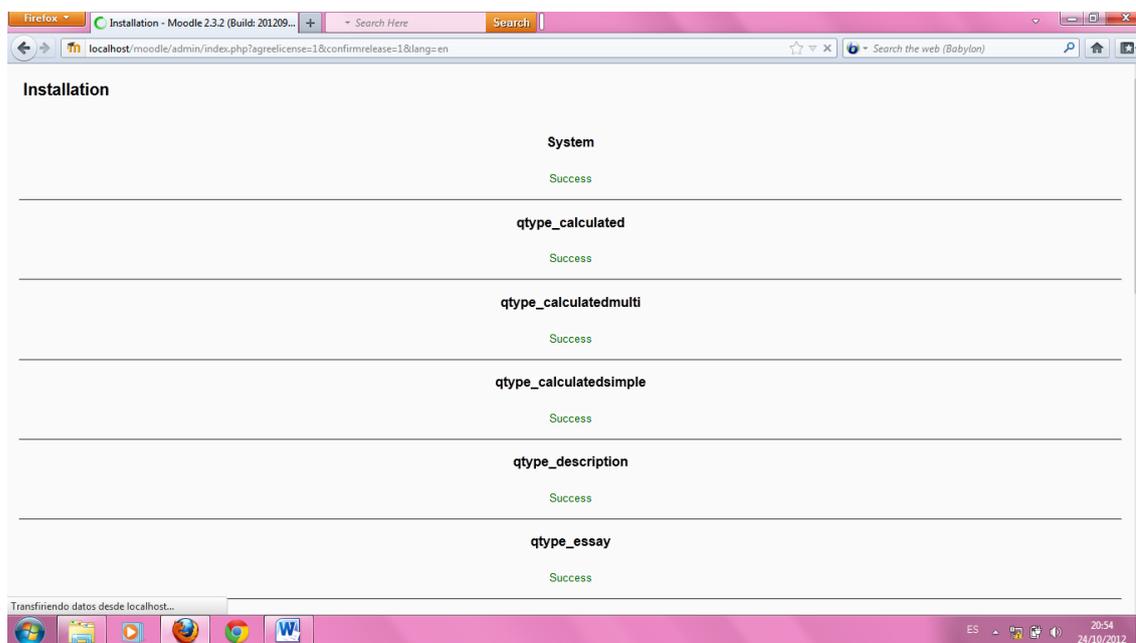


Ilustración XIX: Instalación de Moodle - Paso 7

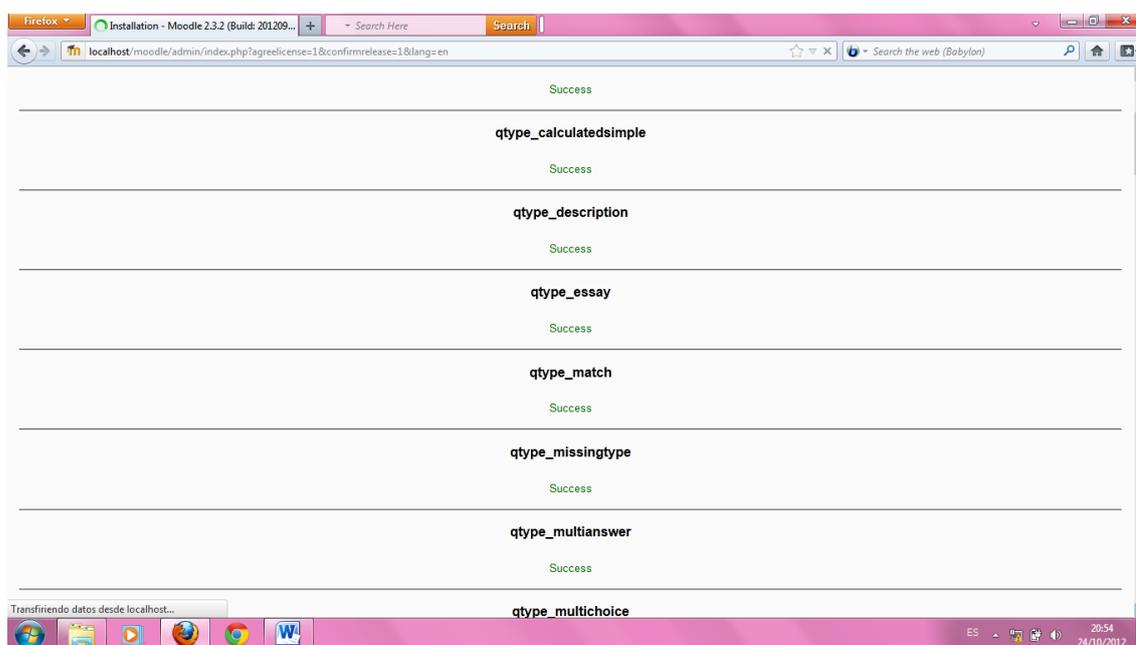


Ilustración XX: Instalación de Moodle - Paso 8

En la siguiente pantalla tendremos que suministrar los datos del administrador de Moodle:

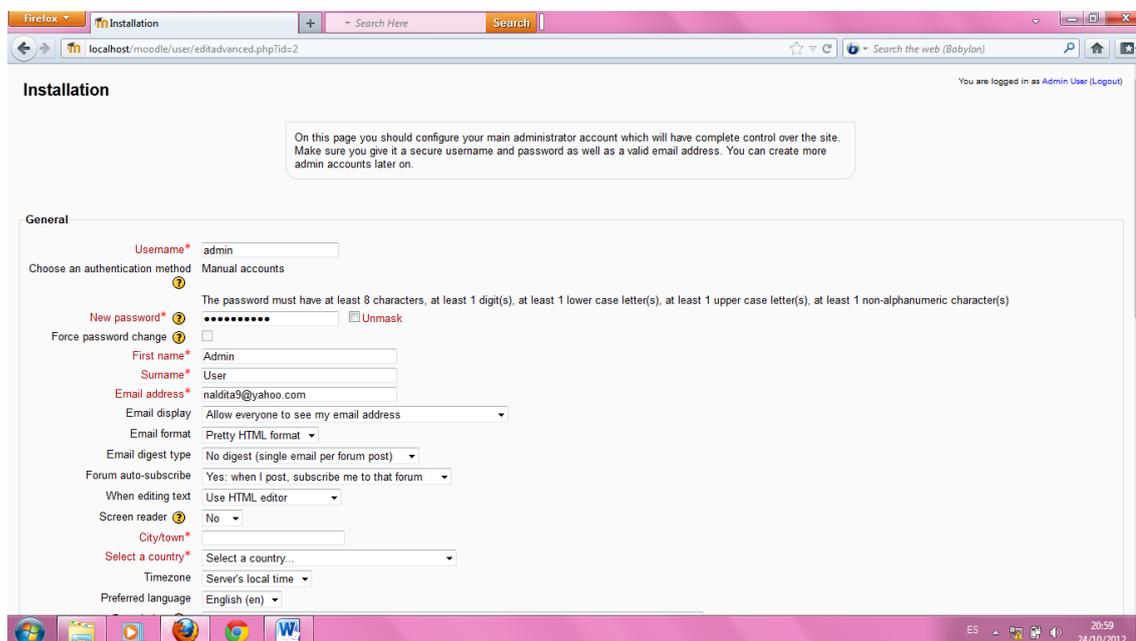


Ilustración XXI: Instalación Moodle - Paso 9

Por último configuramos la página inicial de nuestra plataforma:

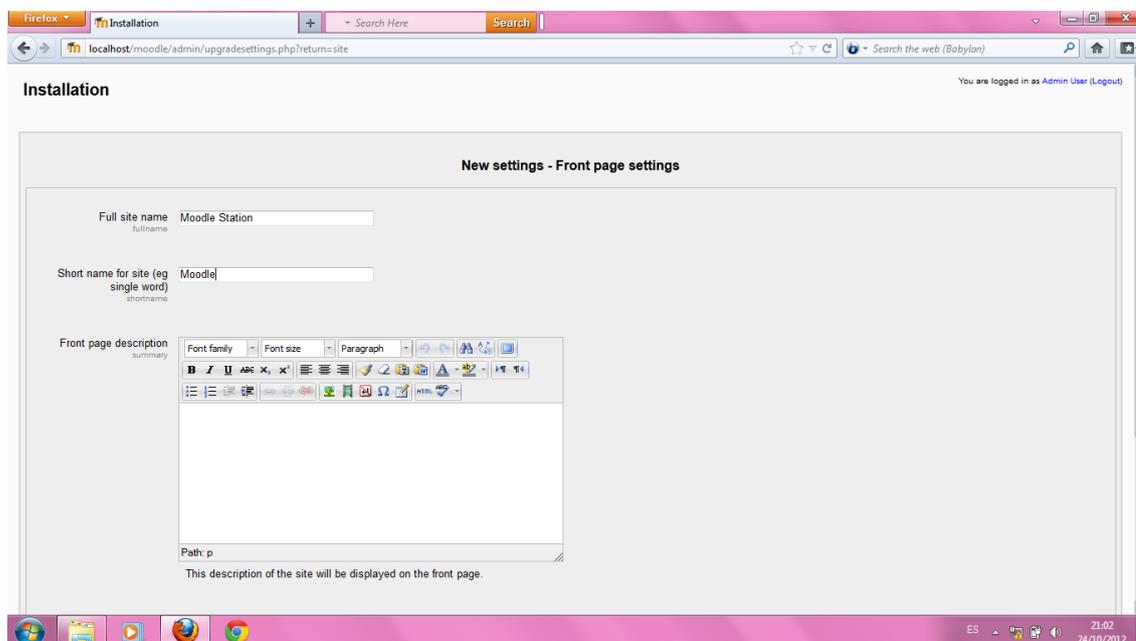


Ilustración XXII: Instalación Moodle - Paso 10

Y, si todo ha ido bien, ya hemos terminado nuestra instalación de Moodle.

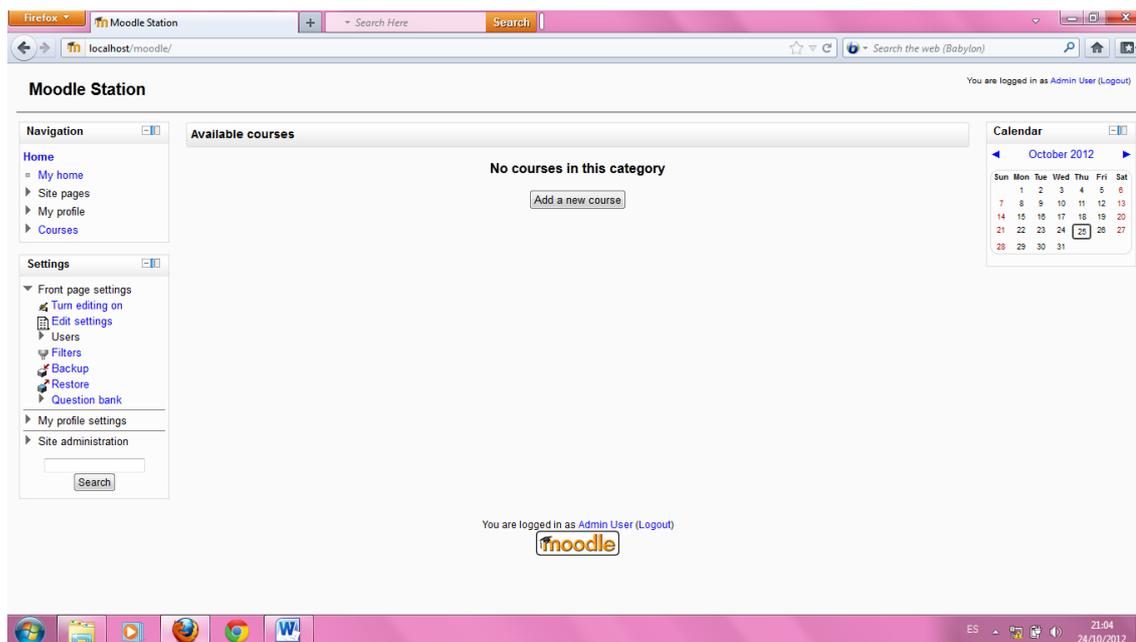


Ilustración XXIII: Moodle instalado correctamente

## ANEXO 2:

### Instalación de Dokeos

Los pasos que aquí se detallan servirán para configurar e instalar satisfactoriamente de Dokeos en el equipo. Estos pasos están pensados para todos aquellos usuarios que utilicen la plataforma para desarrollar módulos.

Se explicará en detalle el proceso de instalación tanto de Dokeos 1.8.5 como de sus requisitos previos en un servidor Windows, en concreto en Windows 7.

### Requisitos Iniciales

Los requerimientos de instalación de Dokeos son los siguientes:

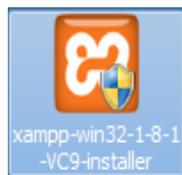
1. **Un servidor web:** La mayoría de los usuarios utilizan Apache.

2. **Una instalación de PHP en funcionamiento:** PHP 5.4 indispensable para la instalación de Dokeos 1.8.5.
3. **Una base de datos:** MySQL, que está completamente soportadas y recomendadas para su uso con Dokeos.

### **Instalación de XAMPP 1.8.1**

Para la Instalación de los requisitos previos a la instalación de Dokeos, es necesario la descarga de un paquete que contiene un Servidor Apache, PHP y MySQL, el cual es XAMPP 1.8.1.

Una vez descargado XAMPP 1.8.1 procedemos a instalarlo dando doble clic en el archivo xampp-win32-1-8-1-VC9-installer.



**Ilustración XXIV: Instalador XAMPP**

Seleccionamos el idioma Inglés y presionamos la opción OK



**Ilustración XXV: Instalación XAMPP (Paso 1)**

Presionamos la opción Next



Ilustración XXVI: Instalación XAMPP (Paso 2)

Seleccionamos los componentes que deseamos que se instalen, Seleccionamos todas las opciones.

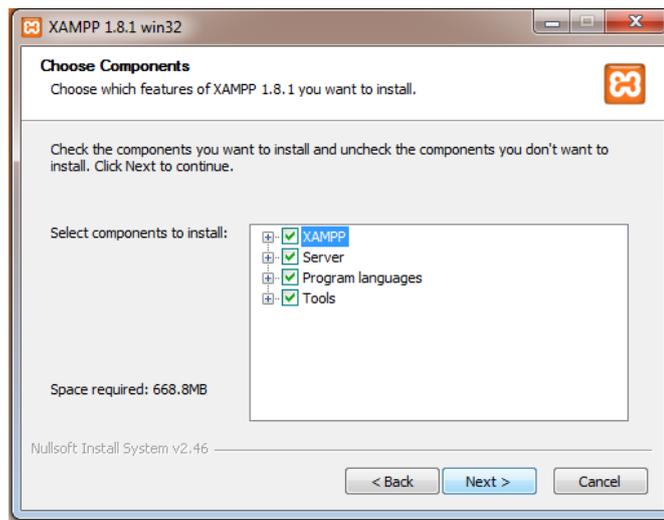
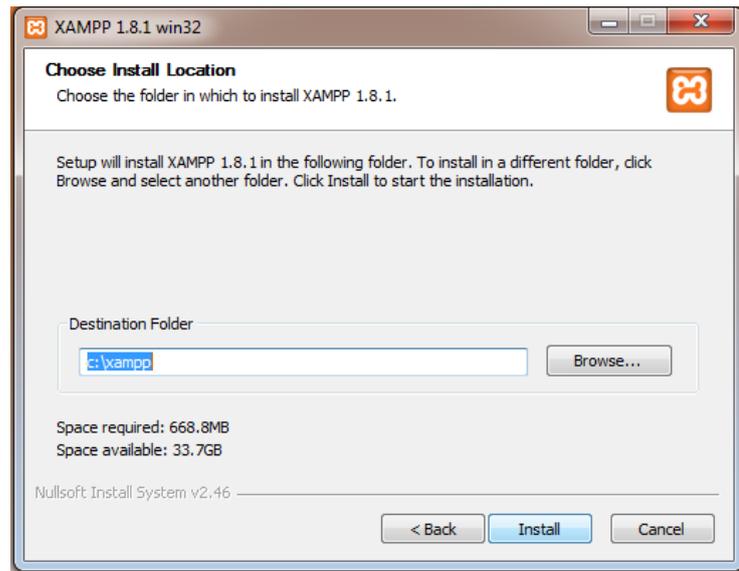


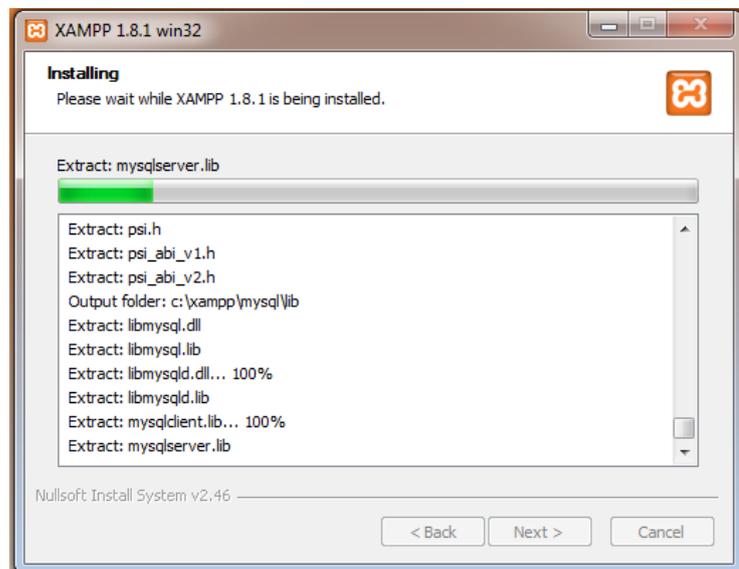
Ilustración XXVII: Instalación XAMPP (Paso 3)

Seleccionamos la ruta donde se instalara XAMPP en nuestro caso en C:\xampp



**Ilustración XXVIII: Instalación XAMPP (Paso 4)**

Posteriormente se procede a extraer los archivos e instalar XAMPP y presionamos la opción Finish.



**Ilustración XXIX: Instalación XAMPP (Paso 5)**

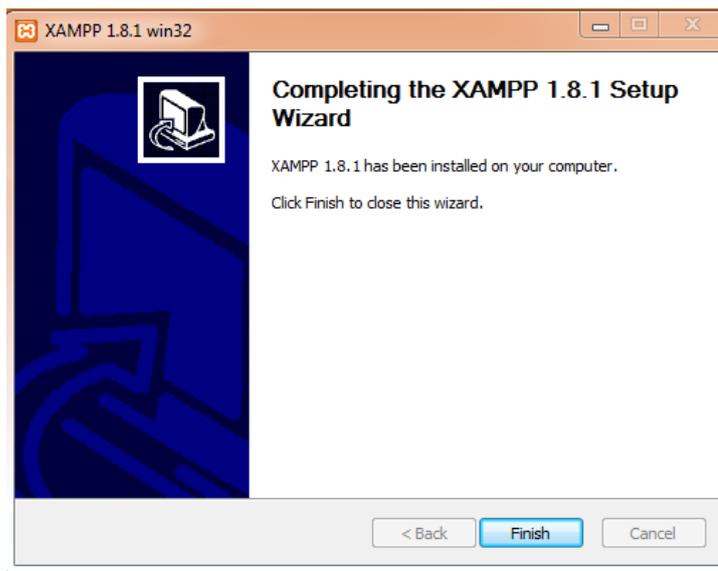


Ilustración XXX: Instalación XAMPP (Paso 6)

Finalmente tenemos instalado XAMPP, aquí la consola de configuración de XAMPP. En la cual iniciamos el Servidor apache y MySQL.

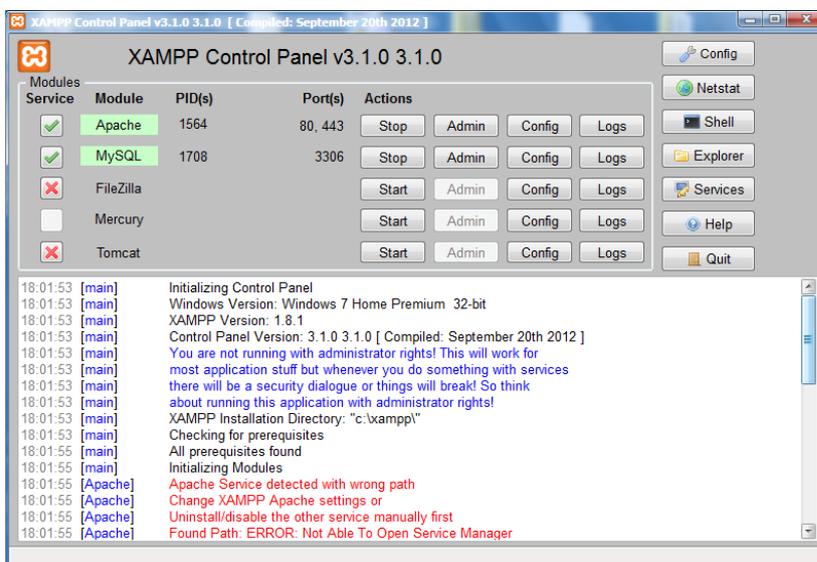


Ilustración XXXI: Consola de configuración de XAMPP

Accedemos al navegador escribiendo localhost, y tenemos instalado correctamente nuestro XAMPP.



Ilustración XXXII: XAMPP instalado correctamente

## Instalación Dokeos

Para instalar Dokeos es necesario descargarnos Dokeos 1.8.5, de su página oficial [www.dokeos.com](http://www.dokeos.com). Posteriormente lo descomprimos en C:\xampp\htdocs\Dokeos.

Luego accedemos al navegador y colocamos <http://localhost/dokeos> y empieza la instalación de Dokeos.

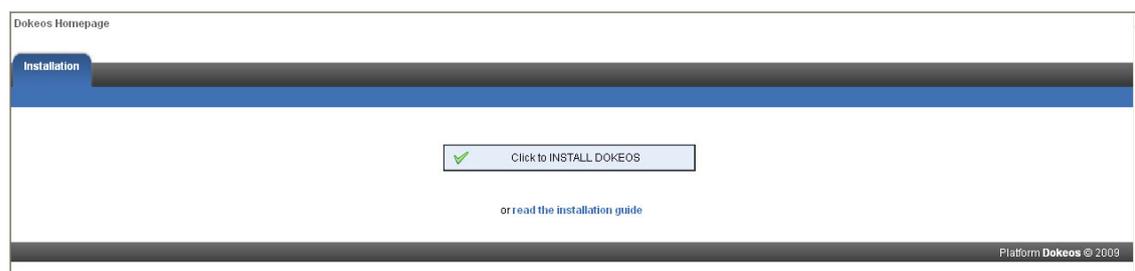


Ilustración XXXIII: Instalación Dokeos - Paso1

Instalar el Dokeos, para eso hay diversos requisitos que configurar como:  
Seleccionamos el idioma y presionamos siguiente.



Ilustración XXXIV: Instalación Dokeos - Paso2

Verificar que tenga todo los requisitos previos, y presionamos nueva instalación, luego aparecerá la licencia, acepte.

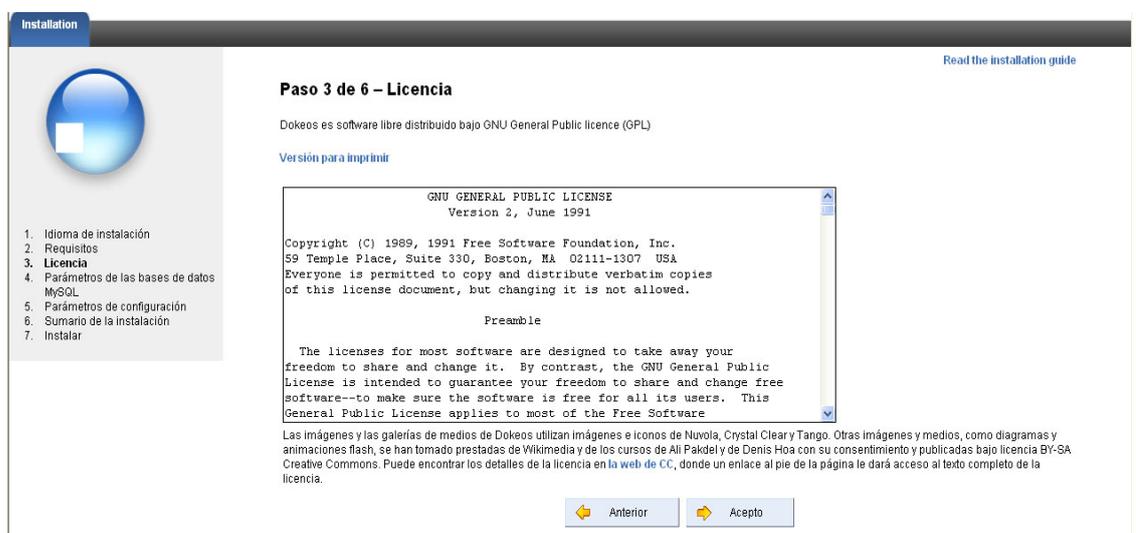


Ilustración XXXV: Instalación Dokeos - Paso 3

Ingresamos los parámetros de MySQL y comprobamos la conexión, presionamos siguiente y colocar el resto de los datos como El servidor de base de datos: localhost, Nombre de usuario de la base de datos por defecto root, Contraseña

Installation

Read the installation guide

### Paso 4 de 6 – Parámetros de las bases de datos MySQL

El script de instalación creará las principales bases de datos de Dokeos. Por favor, recuerde que Dokeos necesitará crear varias bases de datos. Si sólo puede tener una base de datos en su proveedor, Dokeos no funcionará.

Servidor de base de datos: localhost ej. localhost  
Nombre de usuario de la base de datos: root ej. root  
Contraseña de la base de datos: [oculto] ej. aa0ScyTg  
Prefijo MySQL: [ ] No completar si no se requiere

[Parámetros opcionales](#)

Comprobar la conexión con la base de datos

```
MySQL host info: Localhost via UNIX socket
MySQL server version: 5.0.75-Dubuntu10
MySQL protocol version: 10
```

Anterior Siguiente

Ilustración XXXVI: Instalación Dokeos - Paso 4

Llenamos los Parámetros de configuración. Presionamos siguiente.

Installation

Read the installation guide

### Paso 5 de 6 – Parámetros de configuración

Los siguientes valores se grabarán en su archivo de configuración `main/inc/conf/configuration.php`

Idioma principal: spanish

URL de Dokeos (campo obligatorio): [ ]

E-mail del administrador: [ ]

Nombre del administrador: admin

Apellidos del administrador: admin

Teléfono del administrador: [ ]

Nombre de usuario del administrador: admin

Contraseña del administrador (puede que desee cambiarla): admin

Nombre de su plataforma: Campus Virtual

Acrónimo de la organización: [ ]

URL de la organización: [ ]

Método de encriptación:  md5  sha1  Nada

Permitir que los propios usuarios puedan registrarse:  Sí  Recomendado  No

Permitir que los propios usuarios puedan registrarse como creadores de cursos:  Sí  No

Anterior Siguiente

Ilustración XXXVII: Instalación Dokeos - Paso 5

Verificar si todo está correcto y presioné instalar.

Installation Read the installation guide



1. Idioma de instalación
2. Requisitos
3. Licencia
4. Parámetros de las bases de datos MySQL
5. Parámetros de configuración
6. **Sumario de la instalación**
7. Instalar

### Paso 6 de 6 – Última comprobación antes de instalar

Éstos son los valores que ha introducido  
**Imprima esta página para recordar su contraseña y otras configuraciones**

Idioma principal : spanish

Servidor de base de datos : localhost  
Nombre de usuario de la base de datos : root  
Contraseña de la base de datos : \*\*\*\*\*  
Prefijo MySQL : campus\_  
Base de datos principal de Dokeos (BD) : **dokeos\_main** (lea la advertencia inferior)  
Base de datos de seguimiento. Úsela solo si hay varias bases de datos. : **dokeos\_stats** (lea la advertencia inferior)  
Base de datos de usuarios : **dokeos\_user** (lea la advertencia inferior)  
Permitir seguimiento : Si  
Usar Dokeos con una o varias bases de datos : Varios

Permitir que los propios usuarios puedan registrarse : No  
Método de encriptación : md5

E-mail del administrador :  
Nombre del administrador : admin  
Apellidos del administrador : admin  
Teléfono del administrador :  
Nombre de usuario del administrador : **admin**  
Contraseña del administrador (puede que desee cambiarla) : **admin**

Nombre de su plataforma : Campus Virtual  
Acónimo de la organización : Universidad Peruana de las Americas  
URL de la organización : <http://www.ulasamericas.edu.pe>  
URL de Dokeos :

**Atención !**  
El programa de instalación borrará todas las tablas de las bases de datos seleccionadas. Le recomendamos encarecidamente que realice una copia de seguridad completa de todas ellas antes de confirmar este último paso de la instalación.

← Anterior✓ Instalar Dokeos

Ilustración XXXVIII: Instalación Dokeos - Paso 6

Terminado esto ya se puede cargar el Dokeos, el mismo que se muestra de la siguiente forma.



Ilustración XXXIX: Dokeos Instalado

### ANEXO 3:

## COMPONENTES MOODLE

### COMPONENTE MIBLOQUE

#### Archivo *mibloque.php*

```
<?php
//require_once('.././config.php');
class block_mibloque extends block_base{
    function init(){
        $this->title = get_string('pluginname','block_mibloque');
        $this->version = 2011042200;
    }

function get_content(){
    global $USER, $CFG, $DB, $OUTPUT;
    if ($this->content!== NULL){
        return $this->content;
    }
    if (empty($this->instance)){
        $this->content = "";
        return $this->content;
    }

    $this->content = new stdClass;
    $this->content->items = array();
    $this->content->icons = array();
    $this->content->footer = "";
    $context = get_context_instance(CONTEXT_BLOCK, $this->instance->id);
    $this->content->text.='<table width="140" class="userTable"
bgcolor="#F4F4FF" style="border: 1px solid; border-color:#B9DCFF">';
    $this->content->text.='<tr>';
    $this->content->text.='<td width="30" class="user"
align=center>'.<I>'.<B>'. 'PICTURE' .</B>'.</I>'.</td>';
    // $this->content->text .='<td class="user"
align=center>'.<I>'.<B>'. 'ID' .</B>'.</I>'.</td>';
    $this->content->text .='<td width="110" class="user"
align=center>'.<I>'.<B>'. 'NAME' .</B>'.</I>'.</td>';
    // $this->content->text .='<td class="user"
align=center>'.<I>'.<B>'. 'APELLIDO' .</B>'.</I>'.</td>';
```

```

        $this->content->text .='</tr>';
if (has_capability('block/mibloque:view', $context)) {
    $query = ("SELECT id, firstname, lastname FROM {user} ORDER BY id");
    $params = array();
    if ($users = $DB->get_records_sql($query, $params, 0, 8)){
        foreach($users as $user) {
            $this->content->text .= '<tr>';
            $this->content->text.='<td class="user" align=center>'.$OUTPUT-
            >user_picture($user, array('size'=>20)).'</td>';
            //$this->content->text.='<td class="user" align=left>'.$user-
            >id.'</td>';
            $this->content->text.='<td class="user" align =left>'.$user-
            >firstname.' '.$user->lastname.'</td>';
            // $this->content->text.='<td class="user" align =left>'.$user-
            >lastname.'</td>';
            $this->content->text .='</tr>';
        }
        $this->content->text .='</table>';
    }
}
else{
    $this->content->text.= 'No se encontraron usuarios';
}
}else {
    $this->content="";
}
return $this->content;
}

function instance_allow_config() {
    return true;
}

function preferred_width() {
// El valor preferido está en pixeles
    return 180;
}
}
?>
```

**Archivo *styles.css***

```
.block_mibloque .userTable{
    width: 100%;
    /* padding: 0;
    margin: 0;
        background-color:#C9C;*/
    }
}
```

```
.block_mibloque .userTable td {
    padding: 0px;
    margin: 0;
    /*vertical-align: top;*/
    }
}
```

```
.block_mibloque .user {
    width: 100%;
    border: 1;
    font-family: "Courier New", Courier, monospace;
    /*text-align: left;*/
}
}
```

### ***Archivo `version.php`***

```
<?php
defined('MOODLE_INTERNAL') || die();

$plugin->version = 2011061700;    // The current plugin version (Date:
YYYYMMDDXX)
//$plugin->requires = 2012061700;    // Requires this Moodle version
//$plugin->component = 'block_mibloque; // Full name of the plugin (used for
diagnostics)
?>
```

### ***Archivo `db/access.php`***

```
<?php

$block_mibloque_capabilities = array(

    //valores: a que contexto aplica, nombre de la capacidad,
    //tipo de la capacidad y valores para los roles de legado
    'block/mibloque:configure' => array(
```

```
'captype' => 'write',
'contextlevel' => CONTEXT_BLOCK,
'legacy' => array(
    'admin' => CAP_ALLOW
)
),

    'block/mibloque:view' => array(
'captype' => 'read',
    'contextlevel' => CONTEXT_SYSTEM,
    'legacy' => array(
        'guest' => CAP_PREVENT,
        'student' => CAP_PREVENT,
        'teacher' => CAP_PREVENT,
        'editingteacher' => CAP_PREVENT,
        'coursecreator' => CAP_PREVENT,
        'admin' => CAP_ALLOW
    )
)
);
?>
```

#### ***Archivo db/upgrade.php***

```
<?php
function xmldb_block_mibloque_upgrade($oldversion=0){
    global $CFG, $THEME, $db;

    $result = true;
    return result;
}
?>
```

#### ***Archivo lang/en/mibloque.php***

```
<?php

string['pluginname'] = 'Moodle Users';

?>
ANEXO 4:
```

## COMPONENTE CALCULADORA

### Archivo *Calculadora.php*

```
<?php

require_once('.././config.php');
require_once($CFG->libdir.'/formslib.php');

class block_calculadora extends block_base{

function init(){

    $this->title = get_string('pluginname', 'block_calculadora');
    $this->text = '2011042200';
}

function get_content(){

    if($this->content !== NULL){
        return $this->content;
    }

    if (empty($this->instance)){
        $this->content = "";
        return $this->content;
    }

    $this->content= new stdClass;
    $this->content->text= "";
    $this->content->footer= "";
    $this->content->text.= '<form action="" id="cantidades" method="POST">';
        $this->content->text.='<table width="170" style="border: 1px solid;
border-color:#B9DCFF" height="60" border="0" cellpadding="1" cellspacing="0"
bgcolor="#F2F2F2">';
            $this->content->text.= '<tr>'.<td width="48"
bgcolor="#ECECFF">'.<label class="Label">'.get_string('Ingrese_el_numero1',
'block_calculadora').</label>'.</td>';
                $this->content->text.= '<td width="122">'.<input type="text"
name="cant1" value="" class="TextBox" size="4"/>'.</td>'.</tr>';
                    $this->content->text.= '<tr>'.<td width="48"
bgcolor="#ECECFF">'.<label class="Label"
```

```
size="5">'.get_string('Ingrese_el_numero2', 'block_calculadora').</label>'.</td>'.<td
width= "122">'.<input type="text" name="cant2" value="" class="TextBox"
size="4"/>'.</td>'.</tr>;
    //$this->content->text.= '<tr>'.<td>'.<input type="text" name="cant2"
value="" class="TextBox" size="6"/>'.</td>'.</tr>;
    $this->content->text.='<tr>'.<td>'.<select name="type" size="1"
class="Button">
        <option selected value="sumar">Sumar</option>
        <option value="restar">Restar</option>
        <option value="multiplicar">Multiplicar</option>
        <option value="dividir">Dividir</option>
    </select>'.</td>'.<td>'.<input name="Calcular" type="submit"
value="Calcular" class="Button"/>'.</td>'.</tr>;
    //$this->content->text.='<tr>'.<td>'.<input name="Calcular"
type="submit" value="Calcular" class="Button"/>'.</td>'.</tr>;
    if (($_POST['cant1'] !== " ") && ($_POST['cant2'] !== " ") &&
($_POST['type'] !== " ")){
        $num1 = $_POST['cant1'];
        $num2 = $_POST['cant2'];
        $operador = $_POST['type'];
        switch ($operador){
            case 'sumar':
                $result= $num1+$num2;
                $this->content->text.='<tr bgcolor="#ECECFE">'.<td
align="right" style="color:#006699" class="Label">'.El resultado de:'. $num1.'+'.
'$num2.'.'.<td align="center" style="color:#006699"
class="Label">'. $result.</td>'.</tr>'.</td>;
                break;
            case "restar":
                $result = $num1 - $num2;
                $this->content->text.='<tr bgcolor="#ECECFE">'.<td
align="right" style="color:#006699" class="Label">'.El resultado de:'. $num1.'-
'$num2.'.'.<td align="center" style="color:#006699"
class="Label">'. $result.</td>'.</tr>'.</td>;
                break;
            case "multiplicar":
                $result = $num1 * $num2;
                $this->content->text.='<tr bgcolor="#ECECFE">'.<td
align="right" style="color:#006699" class="Label">'.El resultado
de:'. $num1.'*'. $num2.'.'.<td align="center" style="color:#006699"
class="Label">'. $result.</td>'.</tr>'.</td>;
                break;
```

```
        case "dividir":
            $result = $num1 / $num2;
            $this->content->text.='<tr bgcolor="#ECECFE" >'.<td
align="right" style="color:#006699" class="Label">'.El resultado
de:'. $num1.'/'. $num2.'='.<td align="center" style="color:#006699"
class="Label">'. $result.'</td>'.</tr>'.</td>';
            break;
        }
        //$this->content->text.='<tr width="70"
bgcolor="#ECECFE" >'.<td>'.El resultado es:'.</td>'.<td>'. $result.'</td>'.</tr>';
    }
    else{
        echo "ingrese los numeros";
    }
    //$this->content->text.= 'El resultado es:'. calcular();
    $this->content->text.='</table>';
    $this->content->text.='</form>';
    return $this->content;
}
function instance_allow_config() {
    return true;
}
}
?>
```

### **Archivo *version.php***

```
<?php
defined('MOODLE_INTERNAL') || die();

$plugin->version = 2011061700;    // The current plugin version (Date:
YYYYMMDDXX)
//$plugin->requires = 2012061700;    // Requires this Moodle version
//$plugin->component = 'block_mibloque; // Full name of the plugin (used for
diagnostics)
?>
```

### **Archivo *styles.css***

```
.TextBox {
    color: #666666;
    border-color: #004080;
    border: 1px solid;
    font-family: "Comic Sans MS", cursive;
    font-size: 12px;
    background-color: #F8FAFC;
}

.Label {
    color: #006699;
    font-family: "Courier New", Courier, monospace;
    font-size: 12px;
}

.Button {
    color: #006699;
    background-color: #F0F0FF;
    font-weight: bold;
    font-size: 10px;
    font-family: "Comic Sans MS", cursive;
}

.Select {
    color: #006699;
    background-color: #96F;
    border: 1px solid;
    border-color: #004080;
    font-family: "Comic Sans MS", cursive;
}
}
```

### ***Archivo db/upgrade.php***

```
<?php
function xmldb_block_calculadora_upgrade($oldversion=0){
    global $CFG, $THEME, $db;

    $result = true;
    return result;
```

```
}  
?>
```

**Archivo *lang/calculadora.php***

```
<?php
```

```
$string['pluginname'] = 'Calculadora';  
$string['Ingrese_el_numero1'] = 'Numero:';  
$string['Ingrese_el_numero2'] = 'Numero:';  
//$string['El_resultado_es'] = 'the result';
```

```
?>
```

## ANEXO 5:

### COMPONENTE EJEMPLO

#### Archivo *index.php*

```
<?php

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

$id = required_param('id', PARAM_INT); // course

$course = $DB->get_record('course', array('id' => $id), '*', MUST_EXIST);

require_course_login($course);

add_to_log($course->id, 'ejemplo', 'view all', 'index.php?id='.$course->id, '');

$coursecontext = get_context_instance(CONTEXT_COURSE, $course->id);

$PAGE->set_url('/mod/ejemplo/index.php', array('id' => $id));
$PAGE->set_title(format_string($course->fullname));
$PAGE->set_heading(format_string($course->fullname));
$PAGE->set_context($coursecontext);

echo $OUTPUT->header();

if (! $ejemplos = get_all_instances_in_course('ejemplo', $course)) {
    notice(get_string('noejemplo', 'ejemplo'), new moodle_url('/course/view.php',
array('id' => $course->id)));
}

if ($course->format == 'weeks') {
    $table->head = array(get_string('week'), get_string('name'));
    $table->align = array('center', 'left');
} else if ($course->format == 'topics') {
    $table->head = array(get_string('topic'), get_string('name'));
    $table->align = array('center', 'left', 'left', 'left');
} else {
```

```
$table->head = array(get_string('name'));
$table->align = array('left', 'left', 'left');
}

foreach ($ejemplos as $ejemplo) {
    if (!$ejemplo->visible) {
        $link = html_writer::link(
            new moodle_url('/mod/ejemplo.php', array('id' => $ejemplo->coursemodule)),
            format_string($ejemplo->name, true),
            array('class' => 'dimmed'));
    } else {
        $link = html_writer::link(
            new moodle_url('/mod/ejemplo.php', array('id' => $ejemplo->coursemodule)),
            format_string($ejemplo->name, true));
    }

    if ($course->format == 'weeks' or $course->format == 'topics') {
        $table->data[] = array($ejemplo->section, $link);
    } else {
        $table->data[] = array($link);
    }
}

echo $OUTPUT->heading(get_string('modulenameplural', 'ejemplo'), 2);
echo html_writer::table($table);
echo $OUTPUT->footer();

?>
```

### **Archivo lib.php**

```
<?php

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

function ejemplo_supports($feature) {
    switch($feature) {
        case FEATURE_MOD_INTRO:    return true;
        default:                    return null;
    }
}
```

```
function ejemplo_add_instance(stdClass $ejemplo, mod_ejemplo_mod_form $mform =
null) {
    global $DB;

    $ejemplo->timecreated = time();

    # You may have to add extra stuff in here #

    return $DB->insert_record('ejemplo', $ejemplo);
}
```

```
function ejemplo_update_instance(stdClass $ejemplo, mod_ejemplo_mod_form
$mform = null) {
    global $DB;

    $ejemplo->timemodified = time();
    $ejemplo->id = $ejemplo->instance;

    # You may have to add extra stuff in here #

    return $DB->update_record('ejemplo', $ejemplo);
}
```

```
function ejemplo_delete_instance($id) {
    global $DB;

    if (!$ejemplo = $DB->get_record('ejemplo', array('id' => $id))) {
        return false;
    }

    # Delete any dependent records here #

    $DB->delete_records('ejemplo', array('id' => $ejemplo->id));

    return true;
}
```

```
function ejemplo_user_outline($course, $user, $mod, $ejemplo) {

    $return = new stdClass();
    $return->time = 0;
```

```
$return->info = "";
return $return;
}

function ejemplo_user_complete($course, $user, $mod, $ejemplo) {
}

function ejemplo_print_recent_activity($course, $viewfullnames, $timestart) {
    return false; // True if anything was printed, otherwise false
}

function ejemplo_get_recent_mod_activity(&$activities, &$index, $timestart,
    $courseid, $cmid, $userid=0, $groupid=0) {
}
newmodule_get_recent_mod_activity()}

function ejemplo_print_recent_mod_activity($activity, $courseid, $detail, $modnames,
    $viewfullnames) {
}

function ejemplo_cron () {
    return true;
}

function ejemplo_get_extra_capabilities() {
    return array();
}

function ejemplo_scale_used($ejemploid, $scaleid) {
    global $DB;

    if ($scaleid and $DB->record_exists('ejemplo', array('id' => $ejemploid, 'grade' =>
    -$scaleid))) {
        return true;
    } else {
        return false;
    }
}

function ejemplo_scale_used_anywhere($scaleid) {
    global $DB;
```

```
if ($scaleid and $DB->record_exists('ejemplo', array('grade' => -$scaleid))) {
    return true;
} else {
    return false;
}
}

function ejemplo_grade_item_update(stdClass $ejemplo) {
    global $CFG;
    require_once($CFG->libdir.'/gradelib.php');

    $item = array();
    $item['itemname'] = clean_param($ejemplo->name, PARAM_NOTAGS);
    $item['gradetype'] = GRADE_TYPE_VALUE;
    $item['grademax'] = $ejemplo->grade;
    $item['grademin'] = 0;

    grade_update('mod/ejemplo', $ejemplo->course, 'mod', 'ejemplo', $ejemplo->id, 0,
null, $item);
}

function ejemplo_update_grades(stdClass $ejemplo, $userid = 0) {
    global $CFG, $DB;
    require_once($CFG->libdir.'/gradelib.php');

    $grades = array(); // populate array of grade objects indexed by userid

    grade_update('mod/ejemplo', $ejemplo->course, 'mod', 'ejemplo', $ejemplo->id, 0,
$grades);
}

function ejemplo_get_file_areas($course, $cm, $context) {
    return array();
}

function ejemplo_get_file_info($browser, $areas, $course, $cm, $context, $filearea,
$itemid, $filepath, $filename) {
    return null;
}

function ejemplo_pluginfile($course, $cm, $context, $filearea, array $args,
$forcedownload, array $options=array()) {
```

```
global $DB, $CFG;
if ($context->contextlevel != CONTEXT_MODULE) {
    send_file_not_found();
}

require_login($course, true, $cm);

send_file_not_found();
}

function ejemplo_extend_settings_navigation(settings_navigation $settingsnav,
navigation_node $ejemplonode=null) {
}
?>
```

### **Archivo *locallib.php***

```
<?php

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

?>
```

### **Archivo *mod\_form.php***

```
<?php

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

require_once($CFG->dirroot.'/course/moodleform_mod.php');

class mod_ejemplo_mod_form extends moodleform_mod {

public function definition() {

    global $COURSE;
        $mform = $this->_form;

    //-----
    // Adding the "general" fieldset, where all the common settings are showed
    $mform->addElement('header', 'general', get_string('general', 'form'));
}
```

```
// Adding the standard "name" field
$mform->addElement('text', 'name', get_string('newmodulename1', 'ejemplo'),
array('size'=>'64'));
if (!empty($CFG->formatstringstriptags)) {
    $mform->setType('name', PARAM_TEXT);
} else {
    $mform->setType('name', PARAM_CLEAN);
}
$mform->addRule('name', null, 'required', null, 'client');
$mform->addRule('name', get_string('maximumchars', "", 255), 'maxlength', 255,
'client');
$mform->addHelpButton('name', 'newmodulename', 'ejemplo');

// Adding the standard "intro" and "introformat" fields
$this->add_intro_editor();

//-----
$mform->addElement('static', 'label1', 'Nota:', 'Introduzca la descripcion del
modulo!');
$mform->addElement('header', 'Cabecera de ejemplo',
get_string('newmodulefieldset1', 'ejemplo'));
// $mform->addElement('static', 'label2', 'Cabecera de ejemplo', "");
$mform->addElement('text', 'campotexto', get_string('newmodulefieldset',
'ejemplo'), array('size'=>'64'));
//-----
// add standard elements, common to all modules
$this->standard_coursemodule_elements();
//-----
// add standard buttons, common to all modules
$this->add_action_buttons();
}
}
?>
```

### **Archivo *version.php***

```
<?php
```

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

```
$module->version = 0; // If version == 0 then module will not be installed
```

```
$module->version = 2010032200; // The current module version (Date:
YYYYMMDDXX)
$module->requires = 2010031900; // Requires this Moodle version
$module->cron = 0; // Period for cron to check this module (secs)
$module->component = 'mod_ejemplo'; // To check on upgrade, that module sits in
correct place
?>
```

### **Archivo *view.php***

```
<?php
```

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

```
$id = optional_param('id', 0, PARAM_INT); // course_module ID, or
$n = optional_param('n', 0, PARAM_INT); // newmodule instance ID - it should be
named as the first character of the module
```

```
if ($id) {
    $cm = get_coursemodule_from_id('ejemplo', $id, 0, false, MUST_EXIST);
    $course = $DB->get_record('course', array('id' => $cm->course), '*',
MUST_EXIST);
    $ejemplo = $DB->get_record('ejemplo', array('id' => $cm->instance), '*',
MUST_EXIST);
} elseif ($n) {
    $ejemplo = $DB->get_record('ejemplo', array('id' => $n), '*', MUST_EXIST);
    $course = $DB->get_record('course', array('id' => $ejemplo->course), '*',
MUST_EXIST);
    $cm = get_coursemodule_from_instance('ejemplo', $ejemplo->id, $course->id,
false, MUST_EXIST);
} else {
    error('You must specify a course_module ID or an instance ID');
}
require_login($course, true, $cm);
$context = get_context_instance(CONTEXT_MODULE, $cm->id);

add_to_log($course->id, 'ejemplo', 'view', "view.php?id={$cm->id}", $ejemplo->name,
$cm->id);
```

```
/// Print the page header
```

```
$PAGE->set_url('/mod/ejemplo/view.php', array('id' => $cm->id));
$PAGE->set_title(format_string($ejemplo->name));
$PAGE->set_heading(format_string($course->fullname));
$PAGE->set_context($context);

// other things you may want to set - remove if not needed
// $PAGE->set_cacheable(false);
// $PAGE->set_focuscontrol('some-html-id');
// $PAGE->add_body_class('newmodule-'. $somevar);

// Output starts here
echo $OUTPUT->header();

if ($ejemplo->intro) { // Conditions to show the intro can change to look for own
settings or whatever
    echo $OUTPUT->box(format_module_intro('ejemplo', $ejemplo, $cm->id),
'generalbox mod_introbox', 'ejemplointro');
}

// Replace the following lines with you own code
echo $OUTPUT->heading('Yay! It works!');

// Finish the page
echo $OUTPUT->footer();
?>
```

### **Archivo *db/install.xml***

```
<?xml version="1.0" encoding="UTF-8" ?>
<XMLDB PATH="mod/ejemplo/db" VERSION="20101203" COMMENT="XMLDB
file for Moodle mod/ejemplo"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="../../lib/xmldb/xmldb.xsd"
>
<TABLES>
    <TABLE NAME="ejemplo" COMMENT="Ejemplo de prueba">
        <FIELDS>
            <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
UNSIGNED="true" SEQUENCE="true" NEXT="course"/>
```

```
<FIELD NAME="course" TYPE="int" LENGTH="10" NOTNULL="true"
UNSIGNED="true" SEQUENCE="false" COMMENT="Course newmodule activity
belongs to" PREVIOUS="id" NEXT="name"/>
<FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="true"
SEQUENCE="false" COMMENT="name field for moodle instances"
PREVIOUS="course" NEXT="intro"/>
<FIELD NAME="intro" TYPE="text" LENGTH="big" NOTNULL="false"
SEQUENCE="false" COMMENT="General introduction of the newmodule activity"
PREVIOUS="name" NEXT="introformat"/>
<FIELD NAME="introformat" TYPE="int" LENGTH="4" NOTNULL="true"
UNSIGNED="true" DEFAULT="0" SEQUENCE="false" COMMENT="Format of the
intro field (MOODLE, HTML, MARKDOWN...)" PREVIOUS="intro"
NEXT="timecreated"/>
<FIELD NAME="timecreated" TYPE="int" LENGTH="10" NOTNULL="true"
UNSIGNED="true" SEQUENCE="false" PREVIOUS="introformat"
NEXT="timemodified"/>
<FIELD NAME="timemodified" TYPE="int" LENGTH="10" NOTNULL="true"
UNSIGNED="true" DEFAULT="0" SEQUENCE="false"
PREVIOUS="timecreated"/>
</FIELDS>
<KEYS>
<KEY NAME="primary" TYPE="primary" FIELDS="id"/>
</KEYS>
<INDEXES>
<INDEX NAME="course" UNIQUE="false" FIELDS="course"/>
</INDEXES>
</TABLE>
</TABLES>
<STATEMENTS>
<STATEMENT NAME="insert log_display" TYPE="insert" TABLE="log_display"
COMMENT="Initial insert of
records on table log_display. Each record describes how data will be showed by log
reports.">
<SENTENCES>
<SENTENCE TEXT="(module, action, mtable, field) VALUES ('ejemplo', 'add',
'ejemplo', 'name')" /
>
<SENTENCE TEXT="(module, action, mtable, field) VALUES ('ejemplo', 'update',
'ejemplo',
'name')" />
<SENTENCE TEXT="(module, action, mtable, field) VALUES ('ejemplo', 'view',
'ejemplo',
```

```
'name')" />
</SENTENCES>
</STATEMENT>
</STATEMENTS>
</XMLDB>
```

### **Archivo *db/upgrade.php***

```
<?php

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

function xmldb_ejemplo_upgrade($oldversion) {
    global $DB;

    $dbman = $DB->get_manager(); // loads ddl manager and xmldb classes

    if ($oldversion < 2007040100) {

        // Define field course to be added to newmodule
        $table = new xmldb_table('ejemplo');
        $field = new xmldb_field('course', XMLDB_TYPE_INTEGER, '10',
XMLDB_UNSIGNED, XMLDB_NOTNULL, null, '0', 'id');

        // Add field course
        if (!$dbman->field_exists($table, $field)) {
            $dbman->add_field($table, $field);
        }

        // Define field intro to be added to newmodule
        $table = new xmldb_table('ejemplo');
        $field = new xmldb_field('intro', XMLDB_TYPE_TEXT, 'medium', null, null,
null, null, 'name');

        // Add field intro
        if (!$dbman->field_exists($table, $field)) {
            $dbman->add_field($table, $field);
        }

        // Define field introformat to be added to newmodule
        $table = new xmldb_table('ejemplo');
```

```
$field = new xmldb_field('introformat', XMLDB_TYPE_INTEGER, '4',
XMLDB_UNSIGNED, XMLDB_NOTNULL, null, '0',
    'intro');

// Add field introformat
if (!$dbman->field_exists($table, $field)) {
    $dbman->add_field($table, $field);
}
upgrade_mod_savepoint(true, 2007040100, 'ejemplo');
}

if ($oldversion < 2007040101) {

    // Define field timecreated to be added to newmodule
    $table = new xmldb_table('ejemplo');
    $field = new xmldb_field('timecreated', XMLDB_TYPE_INTEGER, '10',
XMLDB_UNSIGNED, XMLDB_NOTNULL, null, '0',
        'introformat');

    // Add field timecreated
    if (!$dbman->field_exists($table, $field)) {
        $dbman->add_field($table, $field);
    }

    // Define field timemodified to be added to newmodule
    $table = new xmldb_table('ejemplo');
    $field = new xmldb_field('timemodified', XMLDB_TYPE_INTEGER, '10',
XMLDB_UNSIGNED, XMLDB_NOTNULL, null, '0',
        'timecreated');

    // Add field timemodified
    if (!$dbman->field_exists($table, $field)) {
        $dbman->add_field($table, $field);
    }

    // Define index course (not unique) to be added to newmodule
    $table = new xmldb_table('ejemplo');
    $index = new xmldb_index('courseindex', XMLDB_INDEX_NOTUNIQUE,
array('course'));

    // Add index to course field
    if (!$dbman->index_exists($table, $index)) {
```

```
        $dbman->add_index($table, $index);
    }

    // Another save point reached
    upgrade_mod_savepoint(true, 2007040101, 'ejemplo');
}

if ($oldversion < 2007040200) {
    upgrade_mod_savepoint(true, 2007040200, 'ejemplo');
}
return true;
}
```

### **Archivo *db/access.php***

```
<?php

defined('MOODLE_INTERNAL') || die();
$capabilities = array(
    'mod/ejemplo:view' => array(
        'captype' => 'read',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'student' => CAP_ALLOW,
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'admin' => CAP_ALLOW
        )
    ),

    'mod/ejemplo:submit' => array(
        'riskbitmask' => RISK_SPAM,
        'captype' => 'write',
        'contextlevel' => CONTEXT_MODULE,
        'legacy' => array(
            'student' => CAP_ALLOW,
            'teacher' => CAP_ALLOW
        )
    ),

    'mod/ejemplo:configure' => array(
        'riskbitmask' => RISK_SPAM,
```

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

### **Archivo lang/ejemplo.php**

```
<?php
defined('MOODLE_INTERNAL') || die();

$string['modulename'] = 'Ejemplo';
$string['modulenameplural'] = 'Ejemplos';
$string['modulename_help'] = 'Use el módulo ejemplo para agregar una a actividad';
$string['newmodulefieldset'] = 'Cabecera';
$string['newmodulefieldset1'] = 'Cabecera del ejemplo';
$string['newmodulename'] = 'Ejemplo';
$string['newmodulename1'] = 'Nombre:';
$string['newmodulename_help'] = 'Este es la ayuda de lo relacionado con el modulo
ejemplo Y la sintaxis que es soportada.';
$string['newmodule'] = 'Ejemplo';
$string['pluginadministration'] = 'Ejemplo administration';
$string['pluginname'] = 'Ejemplo';
?>
```

## ANEXO 6:

### COMPONENTES DOKEOS

#### COMPONENTE DOKEOS\_USER

##### *Archivo index.php*

```
<?php
// name of the language file that needs to be included
$language_file = array('dokeos_user');

// including the global dokeos file
require '../inc/global.inc.php';
$nameTools=get_lang('dokeos_user');
event_access_tool(TOOL_DOKEOS_USER);

define('DOKEOS_USER', true);
Display::display_tool_header();

$table_session_course_user =
Database::get_main_table(TABLE_MAIN_SESSION_COURSE_USER);
    $sql_query = "SELECT DISTINCT user.user_id, ".($is_western_name_order ?
"user.firstname, user.lastname" : "user.lastname, user.firstname").", user.email,
user.official_code
    FROM $table_session_course_user as session_course_user, $table_users as user
    WHERE `course_code` = '".Database::escape_string($course['sysCode'])."'
AND session_course_user.id_user = user.user_id ";

    if ($session_id != 0) {
        $sql_query .= ' AND id_session = '.$session_id;
    }
    $sql_query .= $sort_by_first_name ? ' ORDER BY user.firstname,
user.lastname' : ' ORDER BY user.lastname, user.firstname';
    $rs = Database::query($sql_query, __FILE__, __LINE__);
    while ($user = Database::fetch_array($rs, 'ASSOC')) {
        $data[] = $user;
        //$user_infos = Database::get_user_info_from_id($user['user_id']);
        $a_users[$user['user_id']] = $user;
    }

echo '<div id="content">';
```

```
// Display form
$form->display();
echo '<br />';
$table->display();
echo '</div>';
// secondary actions
echo '<div class="actions"> </div>';
// Display the footer
Display::display_tool_footer();
?>
```

## **ANEXO 7**

### **COMPONENTE DOKEOS\_CALCULATOR**

#### ***Archivo index.php***

```
<?php
// name of the language file that needs to be included
$language_file = array('dokeos_calculator');

$nameTools=get_lang('dokeos_calculator');
event_access_tool(TOOL_DOKEOS_USER);
//$this_section = SECTION_COURSES;

define('DOKEOS_CALCULATOR', true);
Display::display_tool_header();

global $_course, $_user;
//Display::display_header($nameTools, "dokeos_calculator");
require_once(api_get_path(LIBRARY_PATH).'groupmanager.lib.php');

$return = "";
//header('Content-Type: text/html; charset='. $charset);

$result.= '<form action="" id="cantidades" method="POST">';
$result.= '<table width="170" style="border: 1px solid; border-
color:#B9DCFF" height="60" border="0" cellpadding="1" cellspacing="0"
bgcolor="#F2F2F2">';
$result.= '<tr>'.<td width="48"
bgcolor="#ECECFF">'.get_string('Ingrese_el_numero1', 'calculadora').</td>';
```

```
$result.= '<td width="122">'.<input type="text" name="cant1" value=""
size="4"/>'.</td>'.</tr>';
```

```
$result.= '<tr>'.<td width="48"
bgcolor="#ECECFE">'.get_string('Ingrese_el_numero2', 'calculadora').</td>'.<td
width="122">'.<input type="text" name="cant2" value="" size="4"/>'.</td>'.</tr>';
```

```
$result.= '<tr>'.<td>'.<select name="type" size="1" >
<option selected value="sumar">Sumar</option>
<option value="restar">Restar</option>
<option value="multiplicar">Multiplicar</option>
<option value="dividir">Dividir</option>
</select>'.</td>'.<td>'.<input name="Calcular" type="submit"
value="Calcular" />'.</td>'.</tr>';
```

```
if (($_POST['cant1'] !== " ") && ($_POST['cant2'] !== " ") &&
($_POST['type'] !== " ")){
```

```
    $num1 = $_POST['cant1'];
    $num2 = $_POST['cant2'];
    $operador = $_POST['type'];
```

```
    switch ($operador){
    case 'sumar':
```

```
        $resultado= $num1+$num2;
        $result.= '<tr bgcolor="#ECECFE">'.<td align="right"
style="color:#006699" >'.El resultado de:'. $num1.' '+' '. $num2.' '='.<td
align="center" style="color:#006699" >'. $resultado.</td>'.</tr>'.</td>';
        break;
```

```
        case "restar":
        $resultado = $num1 - $num2;
        $result.= '<tr bgcolor="#ECECFE">'.<td align="right"
style="color:#006699" >'.El resultado de:'. $num1.'-'.' $num2.' '='.<td align="center"
style="color:#006699" >'. $resultado.</td>'.</tr>'.</td>';
        break;
```

```
        case "multiplicar":
        $result = $num1 * $num2;
        $result.= '<tr bgcolor="#ECECFE">'.<td align="right"
style="color:#006699" >'.El resultado de:'. $num1.'*'. $num2.' '='.<td align="center"
style="color:#006699" >'. $resultado.</td>'.</tr>'.</td>';
        break;
```

```
        case "dividir":
        $result = $num1 / $num2;
```

```
                $result.='<tr bgcolor="#ECECFF">'.<td align="right"
style="color:#006699" >'.El resultado de:'. $num1.'/'. $num2.'='.<td align="center"
style="color:#006699" >'. $resultado.</td>'.</tr>'.</td>';
                break;
            }
        }
        else{
            echo "ingrese los números";
        }
        $result.='</table>';
        $result.='</form>';
        return $result.;
    }
}
$form->display();
echo '</div>';
// secondary actions
echo '<div class="actions"> </div>';
Display::display_tool_footer();
?>
```

## ANEXO 8:

### ACTIVIDAD EN DOKEOS

```
<?php
// name of the language file that needs to be included
$language_file = array('dokeos_user');

// including the global dokeos file
require './inc/global.inc.php';
$nameTools=get_lang('dokeos_user');
event_access_tool(TOOL_DOKEOS_USER);

define('DOKEOS_USER', true);
Display::display_tool_header();

$table_session_course_user =
Database::get_main_table(TABLE_MAIN_SESSION_COURSE_USER);
    $sql_query = "SELECT DISTINCT user.user_id, ".($is_western_name_order ?
"user.firstname, user.lastname" : "user.lastname, user.firstname").", user.email,
user.official_code
    FROM $table_session_course_user as session_course_user, $table_users as user
    WHERE `course_code` = '".Database::escape_string($_course['sysCode'])."'
AND session_course_user.id_user = user.user_id ";

    if ($session_id != 0) {
        $sql_query .= ' AND id_session = '.$session_id;
    }
    $sql_query .= $sort_by_first_name ? ' ORDER BY user.firstname,
user.lastname' : ' ORDER BY user.lastname, user.firstname';
    $rs = Database::query($sql_query, __FILE__, __LINE__);
    while ($user = Database::fetch_array($rs, 'ASSOC')) {
        $data[] = $user;
        //$user_infos = Database::get_user_info_from_id($user['user_id']);
        $a_users[$user['user_id']] = $user;
    }

echo '<div id="content">';
// Display form
$form->display();
echo '<br />';
$table->display();
echo '</div>';
```

```
// secondary actions
echo '<div class="actions"> </div>';
// Display the footer

    $result.= '<tr>'.<td width="48"
bgcolor="#ECECFF">'.get_string('Ingrese_el_numero1', 'calculadora').</td>';
    $result.= '<td width="122">'.<input type="text" name="cant1" value=""
size="4"/>'.</td>'.</tr>';
    $result.= '<tr>'.<td width="48"
bgcolor="#ECECFF">'.get_string('Ingrese_el_numero2', 'calculadora').</td>'.<td
width= "122">'.<input type="text" name="cant2" value="" size="4"/>'.</td>'.</tr>';

    $result.=<tr>'.<td>'.<select name="type" size="1" >
    <option selected value="sumar">Sumar</option>
    <option value="restar">Restar</option>
    <option value="multiplicar">Multiplicar</option>
    <option value="dividir">Dividir</option>
    </select>'.</td>'.<td>'.<input name="Calcular" type="submit"
value="Calcular" />'.</td>'.</tr>';

    if (($_POST['cant1'] !== " ") && ($_POST['cant2'] !== " ") &&
($_POST['type'] !== " ")){
        $num1 = $_POST['cant1'];
        $num2 = $_POST['cant2'];
        $operador = $_POST['type'];

        switch ($operador){
        case 'sumar':
            $resultado= $num1+$num2;
            $result.=<tr bgcolor="#ECECFF">'.<td align="right"
style="color:#006699" >'.El resultado de:'. $num1.' '+'. $num2.' '='.<td
align="center" style="color:#006699" >'. $resultado.</td>'.</tr>'.</td>';
            break;
        case "restar":
            $resultado = $num1 - $num2;
            $result.=<tr bgcolor="#ECECFF">'.<td align="right"
style="color:#006699" >'.El resultado de:'. $num1.' -'. $num2.' '='.<td align="center"
style="color:#006699" >'. $resultado.</td>'.</tr>'.</td>';
            break;
        case "multiplicar":
            $result = $num1 * $num2;
```

```

        $result.='<tr bgcolor="#ECECFE">'.<td align="right"
style="color:#006699" >'.El resultado de:'. $num1.'*'. $num2.'='.<td align="center"
style="color:#006699" >'. $resultado.</td>'.</tr>'.</td>';
        break;
        case "dividir":
        $result = $num1 / $num2;
        $result.='<tr bgcolor="#ECECFE">'.<td align="right"
style="color:#006699" >'.El resultado de:'. $num1.'/'. $num2.'='.<td align="center"
style="color:#006699" >'. $resultado.</td>'.</tr>'.</td>';
        break;
        }
        }
        else{
        echo "ingrese los números";
        }
        $result.='</table>';
        $result.='</form>';
        return $result.;
        }
    }
    $form->display();
    echo '</div>';
    // secondary actions
    echo '<div class="actions"> </div>';
    Display::display_tool_footer();
    ?>
```

## **ANEXO 9:**

### **MANUAL TÉCNICO MÓDULO ESPOCH**

#### **1. FUNCIONAMIENTO BÁSICO DEL MODULO ESPOCH**

##### **1.1. Introducción al Módulo ESPOCH**

El módulo ESPOCH es un componente desarrollado para dar cumplimiento al nuevo sistema de calificaciones de la Escuela Superior Politécnica de Chimborazo, con el fin de facilitar el trabajo a sus docentes.

El objetivo primordial del Módulo ESPOCH es realizar el cálculo de las diferentes notas de los estudiantes y mostrarlas en base al formato 8-10-10 es decir 28 puntos, repartidos en tres periodos respectivamente.

##### **1.2. Requerimientos**

###### **1.2.1. Requerimientos mínimos del sistema**

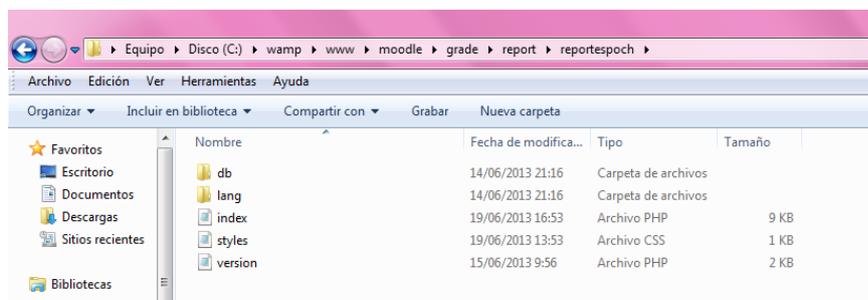
- **Sistema operativo:** Windows 7
- **Procesador:** Intel Pentium4 400MHz
- **Memoria:** 256 MB mínimo

##### **1.3. Desarrollo del Módulo (PHP y MOODLE)**

Para la creación del nuevo módulo orientado al nuevo sistema de calificaciones, se procedió a:

Dentro de este módulo especificaremos las funciones y métodos que nos permitirán acceder a la base de datos y la manipulación de los datos.

Para lo cual crearemos una nueva carpeta con el nombre de **reportespoch** dentro de la carpeta **grade** de Moodle en la siguiente ubicación **C:\wamp\www\moodle\grade\report\reportespoch** dentro de la cual se crearán los archivos:



**Archivos y carpetas creadas para el Módulo ESPOCH**

- **index.php**: que es el archivo donde se programará la estructura del componente. (Ver anexo 10)
- **versión.php**: el cuál especifica la versión actual y la versión requerida para la instalación del componente. (Ver anexo 10)
- **styles.css**: es la hoja de estilo con la que dará la apariencia del componente desarrollado. (Ver anexo 10)

Y las siguientes carpetas:

- La carpeta **reportespoch/lang** en la cual se encuentra la carpeta en dentro de la misma está el archivo:
  - **gradereport\_reportespoch.php** en el cual se encuentran las cadenas del módulo traducidas al idioma en nuestro caso el inglés y así se crearán carpetas de acuerdo a los idiomas en los que se desea mostrar el componente. (Ver anexo 10)
- La carpeta **reportespoch/db** dentro de la carpeta la cual contiene los archivos:

- o **access.php** en el cuál se encuentran las capacidades asignadas a los diferentes roles. (Ver anexo 10)

El archivo **reportesepoch/index.php** incluye las librerías necesarias para el desarrollo del componente.

```
22
23 include_once('../../config.php');
24 require_once($CFG->libdir . '/gradelib.php');
25 require_once $CFG->dirroot.'/grade/lib.php';
26
```

Módulo ESPOCH- código 1

Para el desarrollo del componente es necesario inicializar los aspectos indispensables como el id del curso, requerir que el usuario haya iniciado sesión y las capacidades de mismo, como se muestra en la siguiente figura:

```
26 $courseid = required_param('id', PARAM_INT); // course id
27
28
29
30 // como se mostrará la pagina en el navegador
31 $PAGE->set_url('/grade/report/reportesepoch/index.php', array('id'=>$courseid));
32
33 if (!$course = $DB->get_record('course', array('id' => $courseid))) {
34     print_error('nocourseid');
35 }
36
37 require_login($course);
38 $context = get_context_instance(CONTEXT_COURSE, $course->id);
39
40 require_capability('gradereport/otro:view', $context);
41
42 $report_info = array();
43 $reportesepoch = grade_outcome::fetch_all_available($courseid);
```

Módulo ESPOCH- código 2

Posteriormente se realizan las sentencias SQL necesarias para mostrar los alumnos notas, fechas y promedios, estructurados en tres aportes de 8-10-10. (Ver Anexo 10).

Se realizan los cálculos para el formato definido por la ESPOCH y finalmente se muestra la tabla con los datos calculados según el formato requerido. . (Ver Anexo 10).

```
167 //Imprimimos la tabla del modulo
168 print_grade_page_head($courseid, 'report', 'reportesepoch');
169 echo $html;
170 echo $OUTPUT->footer();
```

Módulo ESPOCH- código 3

## 1.4. Instalación

Para su utilización el nuevo componente, se debe descomprimir el archivo *epoch.zip*, copiar la carpeta *epoch*, dentro de nuestro Moodle, en la siguiente ruta: *moodle/grade/report*.

Ingresar al navegador, y escribir *http://localhost/moodle/*, e ingresar a la plataforma como administrador.



En la pestaña Administración del Sitio, presionamos Notificaciones y nos notificará el componente a instalar.

### Number of plugins requiring your attention: 7

Display the full list of installed plugins

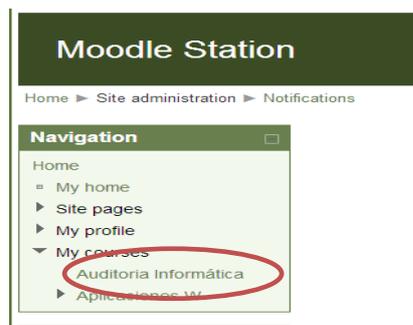
Plugin name	Directory	Source	Current version	New version	Requires	Status
<b>Activity modules</b>						
Ejemplo	/mod/ejemplo	Contributed	2010032200	2010032200	Moodle 2010031900	Installed
<b>Blocks</b>						
Calculadora	/blocks/calculadora	Contributed	2011061700	2011061700		Installed
Configurable Reports	/blocks/configurable_reports	Contributed	2011040103	2011040103	Moodle 2010112400	Installed
Moodle Users	/blocks/mibloque	Contributed	2011061700	2011061700		Installed
Simple Clock	/blocks/simple_clock	Contributed	2011051300	2011051300		Installed
<b>Gradebook reports</b>						
Otro reporte	/grade/report/otro	Contributed	2012061700	2012061700	Moodle 2012061700	Installed
Reporte ESPOCH	/grade/report/reportepoch	Contributed		2012061700	Moodle 2012061700	To be installed

Posteriormente presionamos el boton para la actualizacion en la base de datos y se nos instala el nuevo componente.



Presionamos continuar, e ingresamos como docente a plataforma.

Seleccionamos el curso, del cual es docente en nuestro caso Auditoria Informática.



Seleccionamos la pestaña Calificaciones, de nuestro curso.



En la Selección seleccionamos la opción ESPOCH



Y se nos mostrara las calificaciones de cada estudiante del curso de acuerdo al nuevo sistema de calificaciones de la Escuela Superior Politécnica de Chimborazo. Posteriormente de haber realizado los diferentes trabajos, pruebas y lecciones los estudiantes en el curso virtual.

Nombre/Apellido	Periodo 1 / 8	Periodo 2 / 10	Periodo 3 / 10	Promedio / 28
Geovanny Silva	7.27	3.75	0	11.02
Jimena Villacres	4.36	5	0	9.36
Vicente Carrillo	6.55	2.5	0	9.05
Erika Villacres	6.55	10	0	16.55
Alex Erazo	7.27	0	0	7.27
Jose Luistaxi	3.64	8.75	0	12.39
Rosa Sagñay	0	2.11	0	2.11
Marco Pineda	0	2.11	0	2.11
<b>Promedio General</b>				

## **ANEXO 10:**

### **MODULO ESPOCH**

#### **Archivo *access.php***

***moodle/grade/report/reportespoch/db/access.php***

```
<?php
defined('MOODLE_INTERNAL') || die();

$capabilities = array(

    'gradereport/reportespoch:view' => array(

        'riskbitmask' => RISK_PERSONAL,

        'captype' => 'read',

        'contextlevel' => CONTEXT_COURSE,

        'archetypes' => array(

            'teacher' => CAP_ALLOW,

            'editingteacher' => CAP_ALLOW,

            'manager' => CAP_ALLOW

        )

    )

);
```

#### **Archivo *gradereport\_reportespoch.php***

***moodle/grade/report/reportespoch/lang/en/gradereport\_reportespoch.php***

```
<?php

$string['pluginname'] = 'ESPOCH';

$string['reportespoch:view'] = 'View the outcomes report';
```

#### **Archivo *index.php***

***moodle/grade/report/reportespoch/index.php***

```
<?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/>.
/**
 * The gradebook reportepoch report
 *
 * @package gradereport_repoepoch
 * @copyright 2013 Erika Villacres
 * @license http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
 */
include_once(' ../../config.php');
require_once($CFG->libdir . '/gradelib.php');
require_once $CFG->dirroot.'/grade/lib.php';

$courseid = required_param('id', PARAM_INT); // course id
$PAGE->set_url('/grade/report/reportepoch/index.php', array('id'=>$courseid));

if (!$course = $DB->get_record('course', array('id' => $courseid))) {
```

```
print_error('nocourseid');
}

require_login($course);

$context = get_context_instance(CONTEXT_COURSE, $course->id);

require_capability('gradereport/reportespoch:view', $context);

$report_info = array();

$espoch = grade_outcome::fetch_all_available($courseid);

$html = '<table class="generaltable boxaligncenter" width="50%" cellspacing="1"
cellpadding="5" summary="Espoch Report">' . "\n";

$html .= '<tr align="center">' . '<td style="background-color:#efefff; border-color:#006400;
font-family:Verdana;font-size:12px;font-style:italic;text-align:center;"
>' . '<B>' . 'Nombre/Apellido' . '</B>' . '</td>';

$html .= '<td style="background-color:#efefff; border-color:#006400; font-
family:Verdana;font-size:12px;font-style:italic;text-align:center;" width="70">' . '<B>' . 'Periodo 1
/ 8' . '</B>' . '</td>';

$html .= '<td style="background-color:#efefff; border-color:#006400; font-
family:Verdana;font-size:12px;font-style:italic;text-align:center;" width="70">' . '<B>' . 'Periodo
2 / 10' . '</B>' . '</td>';

$html .= '<td style="background-color:#efefff; border-color:#006400; font-
family:Verdana;font-size:12px;font-style:italic;text-align:center;" width="70">' . '<B>' . 'Periodo
3 / 10' . '</B>' . '</td>';

$html .= '<td style="background-color:#efefff; border-color:#006400; font-
family:Verdana;font-size:12px;font-style:italic;text-align:center;" width="70">' .
'<B>' . 'Promedio / 28' . '</B>' . '</td>';

$html .= '</tr>' . "\n";

$sql = "SELECT u.id, u.firstname, u.lastname
FROM {user} u
JOIN {user_enrolments} ue ON ue.userid = u.id
JOIN {enrol} e ON e.id = ue.enrolid
JOIN {role_assignments} ra ON ra.userid = u.id
WHERE e.courseid = $courseid
AND ra.roleid = 5
AND u.id != 2";
```

```
$params = array();

if ($users = $DB->get_records_sql($sql, $params, 0, 20)){

    $numero = 1;

    foreach($users as $user) {

        $html.= '<tr>';

        $html.='<td style="background-color:#f9f9f9; border-color:#006400; font-
family:Verdana;font-size:12px;font-style:italic" align=left >'. $OUTPUT->user_picture($user,
array('size'=>20));

        $html.=' '.$user->firstname.' '.$user->lastname.'</td>';

        $sql1= "SELECT * FROM {dates}";

        if ($dates = $DB->get_records_sql($sql1, $params, 0, 7)){

            foreach($dates as $date) {

                $fip1 = $date->fip1;

                $ffp1 = $date->ffp1;

                $fip2 = $date->fip2;

                $ffp2 = $date->ffp2;

                $fip3 = $date->fip3;

                $ffp3 = $date->ffp3;

            }

        }

    }

}

$sql1 = "SELECT g.timemodified, g.itemid, g.finalgrade, gi.grademax, SUM(g.finalgrade)as sum,
SUM(gi.grademax) as sumgrademax

        FROM {grade_items} gi

        JOIN {grade_grades} g ON g.itemid = gi.id

        WHERE gi.courseid = $courseid

        AND g.userid = $user->id

        AND gi.itemtype ='mod' AND g.timemodified >= $fip1 AND g.timemodified <=
        $ffp1";

$sql2 = "SELECT g.timemodified, g.itemid, g.finalgrade, gi.grademax, SUM(g.finalgrade)as sum,
SUM(gi.grademax) as sumgrademax

        FROM {grade_items} gi
```

```
JOIN {grade_grades} g ON g.itemid = gi.id
WHERE gi.courseid = $courseid
AND g.userid = $user->id
AND gi.itemtype = 'mod' AND g.timemodified >= $fip2 AND g.timemodified <=
$ffp2";

$sql3 = "SELECT g.timemodified, g.itemid, g.finalgrade, gi.grademax, SUM(g.finalgrade)as sum,
SUM(gi.grademax) as sumgrademax

FROM {grade_items} gi
JOIN {grade_grades} g ON g.itemid = gi.id
WHERE gi.courseid = $courseid
AND g.userid = $user->id
AND gi.itemtype = 'mod' AND g.timemodified >= $fip3 AND g.timemodified <=
$ffp3";

$Total_Periodo = 0;
$totalperiodo1 = 0;
$totalperiodo2 = 0;
$totalperiodo3 = 0;
$total = 0;

if ($items = $DB->get_records_sql($sql1, $params, 0, 60)){
    foreach($items as $item ){
        $total1 = 0;
        $total1 = ( $item->sum * 8)/ $item->sumgrademax;
        $html.= '<td style="background-color:#f9f9f9; border-color:#006400; font-family:
Verdana;font-size:12px;font-style:italic;text-align:center;">'.round($total1, 2).'
```

```
foreach($items as $item ){
$total2 = 0;
$total2= ( $item->sum * 10)/ $item->sumgrademax;
$html.='<td style="background-color:#f9f9f9; border-color:#006400; font-
family:Verdana;font-size:12px;font-style:italic;text-align:center;">'.round($total2, 2).'
```

```
$html.='<td style="background-color:#f9f9f9; border-color:#006400; font-family: Verdana;font-size:12px;font-style:italic;text-align:center;">'.$totalperiodo1.'</td>';
```

```
$html.='<td style="background-color:#f9f9f9; border-color:#006400; font-family: Verdana;font-size:12px;font-style:italic;text-align:center;">'.$totalperiodo2.'</td>';
```

```
$html.='<td style="background-color:#f9f9f9; border-color:#006400; font-family: Verdana;font-size:12px;font-style:italic;text-align:center;">'.$totalperiodo3.'</td>';
```

```
$html.='<td style="background-color:#f9f9f9; border-color:#006400; font-family: Verdana;font-size:12px;font-style:italic;text-align:center;">'.$total.'</td>';
```

```
$html.='<tr>';
```

```
}
```

```
$html.='</table>';
```

```
print_grade_page_head($courseid, 'report', 'reportepoch');
```

```
echo $html;
```

```
echo $OUTPUT->footer();
```

### ***Archivo version.php***

#### ***moodle/grade/report/reportepoch/version.php***

```
<?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/>.
/**
 * Version details for the outcomes report
 *
 * @package gradereport_outcomes
 * @copyright 2013 Erika Villacres
 * @license http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
 */

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

$plugin->version = 2012061700; // The current plugin version (Date: YYYYMMDDXX)
$plugin->requires = 2012061700; // Requires this Moodle version
$plugin->component = 'gradereport_reportespoch'; // Full name of the plugin (used for
diagnostics)
```