

Capítulo I: Análisis del Módulo de seguimiento del uso de la Plataforma MOODLE.

1.1 Introducción.

El presente proyecto tiene como finalidad el seguimiento de cómo es el uso de la plataforma de aprendizaje en línea MOODLE por parte de los profesores de la PUCE sede Quito.

El conocer como se está utilizando la plataforma MOODLE por parte de los docentes es importante ya que con esta información se podrá tomar decisiones para mejorar el uso de la herramienta.

El seguimiento consiste en el análisis y recopilación sistemáticos de información a medida que avanza un proyecto. Su objetivo es mejorar la eficacia y efectividad de un proyecto y organización. Se basa en metas establecidas y actividades planificadas durante las distintas fases del trabajo de planificación. Ayuda a que se siga una línea de trabajo, y además, permite a la administración conocer cuando algo no está funcionando, ya que si se lleva a cabo adecuadamente, es una herramienta de incalculable valor para una buena administración y proporciona la base para la evaluación.

El seguimiento permite determinar si los recursos disponibles son suficientes y están bien administrados, si la capacidad de trabajo es suficiente y adecuada, y si se está haciendo lo que estaba planificado.

1.2 Antecedentes

En los últimos años se ha dado un incremento en el uso de la tecnología para dar apoyo a la enseñanza. Con el apareamiento de la “web2.0”¹ la enseñanza a través del internet se ha convertido en una manera fácil y cómoda de enseñar, ya que no se necesita estar presente en las clases, y se puede revisar el material cada vez que el alumno lo necesite.

En la PUCE se creó en el año 2010 el “Departamento de Nuevas Tecnologías”; el cual es el encargado de capacitar a los docentes de la institución en el uso de la plataforma de “e-learning”² MOODLE³ para que puedan utilizar de forma eficiente y correcta la plataforma como apoyo a las clases presenciales.

En años anteriores, la universidad contaba con algunas plataformas que ayudaban a los docentes a subir materiales al internet para que sus alumnos puedan revisarlos fuera de clases.

¹ Se nombra así a la **tecnología Web** basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los *blogs*, los *wikis* o las *folcsonomías*, que fomentan la colaboración y el intercambio ágil de información entre los usuarios de una comunidad o red social.(Dirección de Informática PUCE)

²Del inglés electronic learning. El e-learning, en su más amplio sentido, puede definirse como todo proceso formativo que implique el uso de las nuevas tecnologías de la información y la comunicación (Davies G. Hewer S., 2008; Gavin Dudeney Nicky Hockly, 2007). Así, este tipo de aprendizaje electrónico implica la distribución de conocimientos a través de Internet, intranets, entornos virtuales, actividades pedagógicas con asistencia computacional, software educativo y tecnologías móviles, entre otros medios. El aprendizaje a distancia (online learning) y la teleformación (e-training) son dos términos asociados con el e-learning, puesto que son tecnologías orientadas a la creación de conocimiento transferible y al desarrollo de habilidades relacionadas con aspectos individuales u organizativos(Terminología de E-learning)

³ Del inglés Modular Object-Oriented Dynamic Learning Environment. En español Entorno de Aprendizaje Dinámico Modular, Orientado a Objetos. Es un sistema de administración de usuarios-estudiantes y de cursos, de distribución libre y basado en el constructivismo y el aprendizaje colaborativo, que permite a los docentes crear entornos educativos en línea. El sitio web de Moodle es <http://moodle.org>.(Terminología de E-learning)

La universidad hizo un estudio acerca de cuál es el uso de las TICs⁴ por parte de los docentes. Este estudio se lo realizó a través de encuestas en la página PuceVirtual. Los datos que se obtuvieron fueron que el 50% de los docentes no utilizan ninguna plataforma para ayuda a sus clases, el 14% utilizaba “Manhattan”⁵, el 23% utilizaba MOODLE, y el 13% utilizaba otros tipos de TICs.

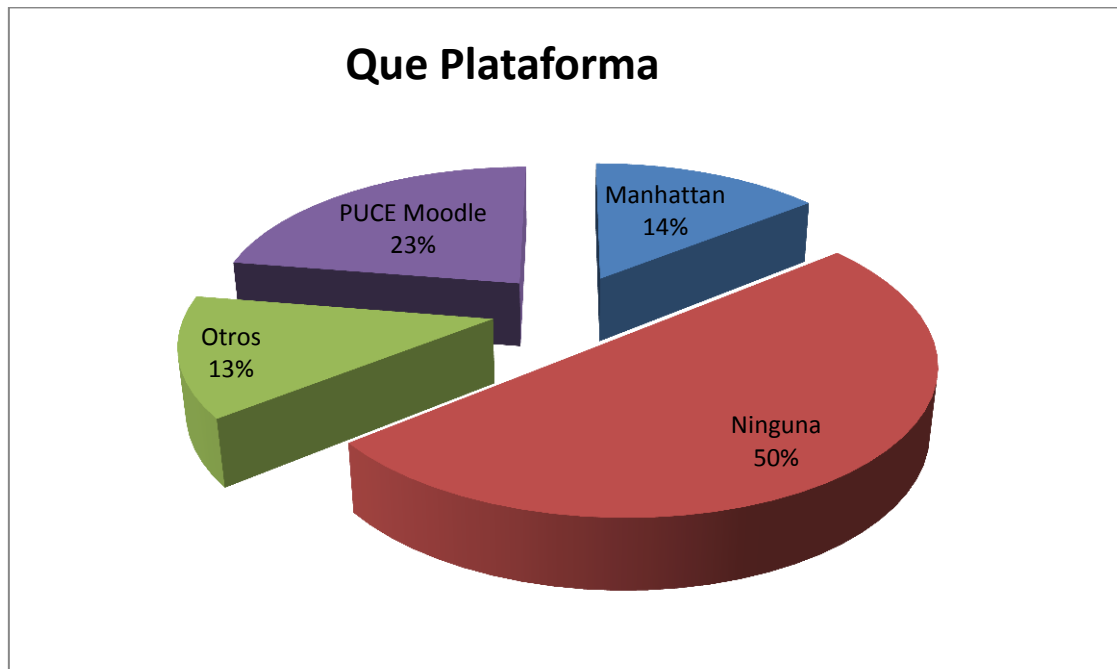


Gráfico 1 Uso de Plataformas por parte de los docentes de la PUCE⁶ Fuente: Departamento de Nuevas Tecnologías

Como vemos sólo el 50% de los docentes utilizan algún tipo de TICs como apoyo a sus clases. Esto se debe principalmente a que el 50% de los docentes tienen más de 50 años, lo cual implica una negativa al uso de la tecnología por varios motivos que no los voy a abordar ya que no es el objetivo de este trabajo. En la actualidad la plataforma Manhattan está fuera de uso, y se ha incentivado el uso de la plataforma MOODLE como plataforma oficial de aprendizaje en línea en la PUCE.

⁴Tecnologías de la Información y de la Comunicación (en inglés se utiliza comúnmente la sigla IT: Information Technology) conjunto de herramientas, habitualmente de naturaleza electrónica, como redes y dispositivos cuyo objetivo es facilitar determinadas tareas. En la actualidad, se utilizan, por ejemplo, una gran variedad de sistemas y medios informáticos para el almacenamiento, tratamiento, difusión y transmisión de la información persiguiendo diferentes finalidades. (Terminología de E-learning)

⁵ Plataforma para subir archivos a un servidor. (N.A)

⁶(Departamento de Nuevas Tecnologías, 2010)

1.2.1 ¿Cómo está actualmente el proyecto PuceMoodle?

El proyecto consta de 3 etapas claramente definidas en el plan operativo que desarrollo el “Departamento de Nuevas Tecnologías” en el año 2009.

“Fase 1: Presencialidad + Apoyo de TICs y EVAs que se utilicen en la PUCE de manera oficial (Los docentes deben presentar en su programación micro curricular el porcentaje de trabajo autónomo que realizarán los estudiantes apoyado en el uso de TICS, para la aprobación del respectivo Consejo de Facultad o Escuela).

Fase 2: *Blended Learning* (Aprendizaje mixto). Una parte del curso debe ser virtual, según políticas definidas exclusivamente por la Dirección General Académica.

Fase 3: E-learning (Todo un curso, Carrera o Programa) de manera virtual, previa la autorización de la instancia correspondiente”⁷.

Actualmente se está en la “Fase 2” del proyecto en donde se da una parte del curso en forma presencial y otra en forma virtual.

Además la plataforma MOODLE de la PUCE consta de 168 cursos agrupados en 27 categorías y consta de 5100 participantes (entre alumnos y docentes).

⁷ Plan para la implementación de la Educación no Presencial en la PUCE (PUCEVirtual)

1.2.2 ¿Cómo se opera actualmente?

1.2.2.1 Tecnología.

Actualmente la plataforma MOODLE corre bajo un servidor Linux⁸ Red Hat 5.

Software.

El software que actualmente está instalado para que funcione MOODLE es Apache 2⁹ y con base de datos MySQL 2¹⁰.

1.2.2.2 Recurso Humano.

El Ing. Marcelo Pérez quien es el encargado de la administración de la plataforma MOODLE de la PUCE.

1.2.3 Fuente del Problema.

La herramienta de e-learning MOODLE fue implantado en la Universidad en el año 2008, con el objetivo de usar esta herramienta como apoyo a las clases presenciales.

Desde ese año hasta el año 2010 se ha estado capacitando al personal docente en el uso de la herramienta. Se ha notado un incremento en el uso y en el buen uso de la herramienta tanto por parte de los docentes como por parte de los estudiantes.

A pesar del notorio incremento en el uso de la herramienta, no existen datos para ver como es el uso de la herramienta. Por esta razón se propone la realización de este proyecto para obtener datos concretos de cómo es el uso de esta herramienta, es decir hacer un seguimiento del trabajo realizado por los docentes en el uso de la plataforma MOODLE.

⁸GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o *kernel* libre similar a Unix denominado **Linux**, que es usado con herramientas de sistema GNU (<http://es.wikipedia.org/wiki/GNU/Linux>)

⁹ El **servidor HTTP Apache** es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1² y la noción de sitio virtual (http://es.wikipedia.org/wiki/Servidor_HTTP_Apache)

¹⁰ **MySQL** es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones (<http://es.wikipedia.org/wiki/MySQL>)

Capítulo II: Herramientas para el desarrollo.

2.1 Herramientas.

Dreamweaver.



Adobe Dreamweaver es una aplicación que está destinada a la construcción y edición de sitios y aplicaciones Web basados en estándares. Fue creado por la empresa Macromedia y actualmente es producida por Adobe Systems. Dreamweaver es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium (W3C)¹¹.

Se ha escogido este Dreamweaver como editor ya que además de soportar PHP y dar soporte a hojas de estilo CSS¹², se puede editar HTML para crear la interfaz gráfica. Este editor es de fácil uso y configuración lo que lo hace una excelente herramienta para la codificación del módulo que se propone realizar.

¹¹ El Consorcio World Wide Web (W3C) es una comunidad internacional donde las organizaciones Miembro [ingles], personal [ingles] a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web [ingles]. Liderado por el inventor de la Web Tim Berners-Lee [ingles] y el Director Ejecutivo (CEO) Jeffrey Jaffe [ingles], la misión del W3C es guiar la Web hacia su máximo potencia (

¹² CSS (Cascading Style Sheets, u Hojas de Estilo en Cascada) es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación (Maestros del Web)

Fussion Charts Free.



Fussion Charts Free es un sistema de gráficas para estadísticas realizado en Flash. Los datos los puede recibir desde un fichero xml o desde algún lenguaje de servidor como PHP, ASP, Java, etc.

La ventaja de esta aplicación es que se la puede insertar en un documento HTML o PHP y se la puede visualizar en cualquier navegador.

Se distribuye bajo licencia GNU por lo que se lo puede utilizar sin costo, se puede copiar y modificar, bajo las condiciones que indica esta licencia.

DBDesigner 4.



DBDesigner 4 es una herramienta visual de trabajo con base de datos. Proporciona facilidades para el diseño, el modelado, la creación y el mantenimiento de bases de datos.

Soporta multitud de opciones, entre las que se encuentran:

- Modo de diseño o consulta.
- Ingeniería inversa de bases de datos MySQL, Oracle, MSSQL y cualquiera con driver ODBC.

- Control de versiones.
- Constructor de queries.
- Soporte especial para MySQL.
- Producto gratuito bajo licencia GPL

Se va utilizar esta herramienta para realizar la reingeniería de la base de datos de MOODLE y ver cómo están relacionadas las tablas.

2.1.2 PHP



“PHP, acrónimo de "*PHP: Hypertext Preprocessor*", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP” (Wikipedia, 2011).

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Características de PHP.

Ventajas:

- Es un lenguaje multiplataforma.
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

Desventajas:

- Como es un lenguaje que se interpreta en ejecución, no existe la manera de hacer un debug¹³ para ver si la codificación tiene errores de sintaxis.
- Al ser un lenguaje que se interpreta, en el momento que existe un error de carácter lógico, la ejecución del programa simplemente se detiene y no se sabe en qué línea de código está el error para poder corregirlo.
- Al ser un lenguaje flexible cada programador utiliza su propio estilo de programar, lo que lo hace difícil entender el código de otros programadores.

2.1.4 MySql.



MySQL es la base de datos open source¹⁴ más popular. Su continuo desarrollo y su creciente popularidad hace de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.

¹³**Depuración de programas** es el proceso de identificar y corregir errores de programación. (<http://es.wikipedia.org/wiki/Debug>)

¹⁴**Código abierto** es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre. (http://es.wikipedia.org/wiki/C%C3%B3digo_abierto)

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

2.2 Modelo de Desarrollo.

2.2.1 Introducción.

“Un modelo es una representación abstracta, conceptual, gráfica o visual de fenómenos, sistemas o procesos a fin de analizar, describir, explicar, simular, explorar, controlar y predecir fenómenos o procesos. Un modelo permite determinar un resultado final a partir de unos datos de entrada.” (Wikipedia, 2011)

Un modelo de desarrollo de software establece el orden en el que se harán las cosas en el proyecto. Nos provee de requisitos de entrada y salida para cada una de las actividades. El modelo de desarrollo de software nos ayuda a la forma en la que vamos a construir el producto.

Un modelo de desarrollo tiene un ciclo de vida¹⁵ que nos ayudará a controlar las actividades del proyecto desde el inicio al fin del mismo, mientras que el modelo de desarrollo nos ayuda a la forma en que vamos a construir el producto; por lo que ambos se complementan para generar un producto de alta calidad.

¹⁵ El término **ciclo de vida del software** describe el desarrollo de software, desde la fase inicial hasta la fase final (<http://es.kioskea.net/contents/genie-logiciel/cycle-de-vie.php3>)

Modelos de Desarrollo de Software.

- Modelo en Cascada.
- Modelo de Prototipos.
- Modelo en Espiral.
- Desarrollo por Etapas.
- Desarrollo iterativo y creciente o Iterativo Incremental.
- RAD.
- Desarrollo Concurrente.
- Proceso Unificado.
- RUP (Proceso Unificado Racional).

Para el desarrollo del proyecto dadas las características del mismo y en base a las características de los modelos expuestos anteriormente he decidido utilizar el modelo “Iterativo Incremental”.

2.2.2 Desarrollo Iterativo Incremental.

El modelo Iterativo Incremental es un modelo de tipo evolutivo, es decir los requisitos van a ir cambiando en el transcurso del proceso. Este modelo es aplicable cuando los requisitos no están completamente definidos y van a cambiar.

El modelo es aconsejable para el desarrollo de software en el cual se observe, en su etapa inicial de análisis, que posee áreas bastante bien definidas a cubrir, con suficiente independencia como para ser desarrolladas en etapas sucesivas. Estas áreas a cubrir suelen

tener distintos grados de prioridad por lo cual las mismas se deben priorizar en un análisis previo, es decir, definir cuál será la primera, la segunda, y así sucesivamente.

En el modelo incremental se procede a desarrollar módulos con entregas parciales de los mismos. Los módulos son entregados de acuerdo a la prioridad que se les dio en la etapa de análisis.

Este modelo brinda flexibilidad para que durante el desarrollo se incluyan cambios en los requisitos por parte del usuario. Un cambio de requisitos propuesto y aprobado puede analizarse e implementarse como un nuevo incremento o, eventualmente, podrá constituir una mejora/adequación de uno ya planeado.

El modelo iterativo incremental no obliga a especificar con precisión y detalle absolutamente todo lo que el sistema debe hacer antes de ser construido (como el caso del cascada, con requisitos ya definidos). Sólo se hace en el incremento en desarrollo. Esto torna más manejable el proceso y reduce el impacto en los costos ya que en caso de alterar o rehacer los requisitos, solo afecta una parte del sistema. Aunque, lógicamente, esta situación se agrava si se presenta en estado avanzado, es decir en los últimos incrementos.

En definitiva, el modelo facilita la incorporación de nuevos requisitos durante el desarrollo.

Ciclo de Vida.

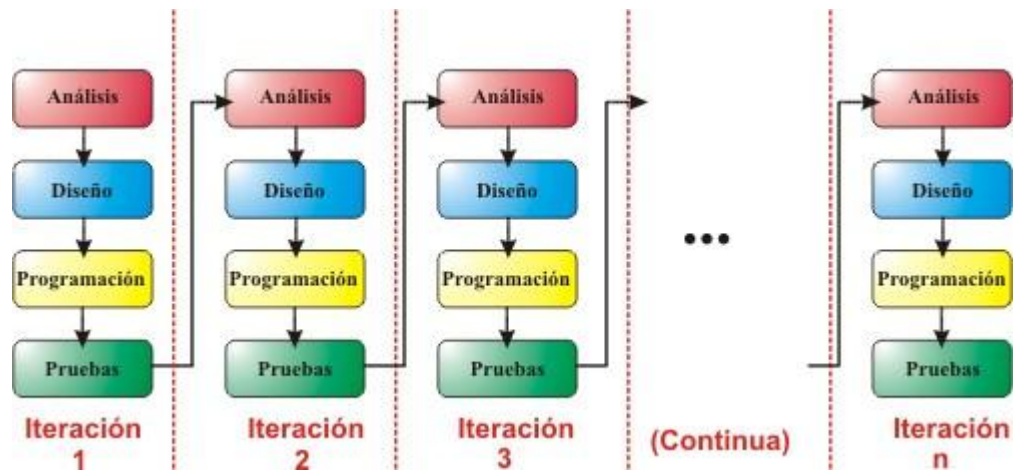


Gráfico 2 1 Ciclo de Vida Modelo Iterativo Incremental Fuente: <http://xherrera334.blogspot.es>

La idea principal detrás de mejoramiento iterativo es desarrollar un sistema de programas de manera incremental, permitiéndole al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema. El aprendizaje viene de dos vertientes: el desarrollo del sistema, y su uso (mientras sea posible). Los pasos claves en el proceso son comenzar con una implementación simple de los requerimientos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

El proceso en sí mismo consiste de:

- Etapa de inicialización.
- Etapa de Iteración.
- Lista de control de proyecto.

Etapa de Inicialización.

Se crea una primera versión del sistema. El objetivo de esta etapa es crear un producto con el que el usuario pueda interactuar para retroalimentar el proceso. Debe ofrecer una muestra de los aspectos principales del problema y proveer una solución simple para ser implementada fácilmente. Para guiar el proceso de iteración se crea una lista de control de proyecto, que contiene un historial de todas las tareas que necesitan ser realizadas. Incluye cosas como nuevas funcionalidades para ser implementadas, y áreas de rediseño de la solución ya existente. Esta lista de control se revisa periódica y constantemente como resultado de la fase de análisis.

Etapa de Iteración.

Esta etapa involucra el rediseño e implementación de una tarea de la lista de control de proyecto, y el análisis de la versión más reciente del sistema. El objetivo del diseño e implementación de cualquier iteración es ser simple, directa y modular, para poder soportar el rediseño de la etapa o como una tarea añadida a la lista de control de proyecto. El código puede representar la mayor fuente de documentación del sistema.

El análisis de una iteración se basa en la retroalimentación del usuario y en el análisis de las funcionalidades disponibles del programa. Involucra el análisis de la estructura, modularidad, usabilidad, confiabilidad, eficiencia y eficacia (alcanzar las metas). La lista de control del proyecto se modifica bajo la luz de los resultados del análisis.

Las guías primarias que guían la implementación y el análisis incluyen:

- Cualquier dificultad en el diseño, codificación y prueba de una modificación debería apuntar a la necesidad de rediseñar o recodificar.
- Las modificaciones deben ajustarse fácilmente a los módulos fáciles de encontrar y a los aislados. Si no es así, entonces se requiere algún grado de rediseño.
- Las modificaciones a las tablas deben ser especialmente fáciles de realizar. Si dicha modificación no ocurre rápidamente, se debe aplicar algo de rediseño.
- Las modificaciones deben ser más fáciles de hacer conforme avanzan las iteraciones. Si no es así, hay un problema primordial usualmente encontrado en un diseño débil o en la proliferación excesiva de parches al sistema.
- Los parches normalmente deben permanecer solo por una o dos iteraciones. Se hacen necesarios para evitar el rediseño durante una fase de implementación.
- La implementación existente debe ser analizada frecuentemente para determinar que tan bien se ajusta a las metas del proyecto.
- Las facilidades para analizar el programa deben ser utilizadas cada vez para ayudar en el análisis de implementaciones parciales.
- La opinión del usuario debe ser solicitada y analizada para indicar deficiencias en la implementación referida por él.

Desventajas:

Requiere de un cliente involucrado durante todo el curso del proyecto. Hay clientes que simplemente no estarán dispuestos a invertir el tiempo necesario.

Infunde responsabilidad en el equipo de desarrollo al trabajar directamente con el cliente, requiriendo de profesionales sobre el promedio.

2.3 ¿Qué es MOODLE?

2.3.1 Introducción.

MOODLE es un Ambiente Educativo Virtual, sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conoce como LMS (Learning Management System).

Moodle fue creado por Martin Dougiamas, quien fue administrador de WebCT en la Universidad Tecnológica de Curtin. Basó su diseño en las ideas del constructivismo¹⁶ en pedagogía que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas y en el aprendizaje colaborativo.

La idea es que un profesor que utiliza las ideas del constructivismo crea un ambiente centrado en el estudiante. Al crear el ambiente en torno al estudiante le ayuda a construir el

¹⁶ Postula la necesidad de entregar al alumno herramientas que le permitan crear sus propios procedimientos para resolver una situación problemática, lo cual implica que sus ideas se modifiquen y siga aprendiendo(http://es.wikipedia.org/wiki/Constructivismo_%28pedagog%C3%ADa%29)

conocimiento en base a sus habilidades y conocimientos propios en lugar de dar información que se supone que los estudiantes deben conocer.

La primera versión de la herramienta apareció el 20 de agosto de 2002 y, a partir de allí han aparecido nuevas versiones de forma regular.

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:

- Su diseño es modular, permitiendo gran flexibilidad para agregar y suprimir funcionalidades en muchos niveles.
- Se ejecuta sin necesidad de cambios en el sistema operativo bajo Unix, Linux, Windows, Mac OS X, Netware y todos aquellos sistemas operativos que soporten PHP.
- Soporta las principales marcas de manejadores de bases de datos.
- Dispone de un sistema interno capaz de reparar y actualizar su base de datos cada cierto tiempo.

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

- Promueve una pedagogía constructivista social. Dado el carácter colaborativo de las herramientas utilizadas en él y la filosofía de trabajo en la que se sustenta.
- Cuenta con una interfaz atractiva, de tecnología sencilla, ligera, eficiente y compatible.
- Permite niveles de acceso a los cursos.
- Permite la clasificación de los contenidos de fácilmente, lo que permite un ordenamiento lógico.
- Es fácil de instalar, por lo que no precisa un nivel avanzado de conocimientos informáticos para poder instalarlo.

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.

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 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.

2.3.2 Estructura de MOODLE.

MOODLE está estructurado de tal forma que la organización sea de fácil manejo. La estructura de MOODLE se hace de forma jerárquica.

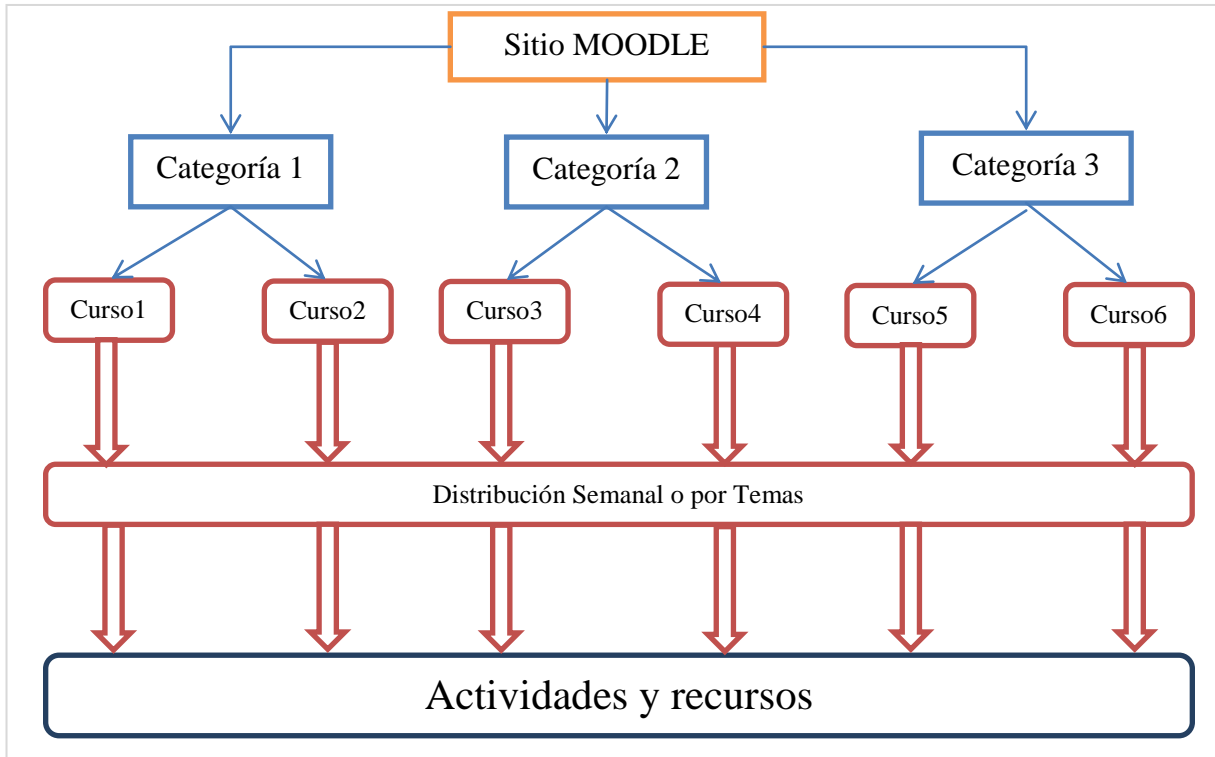


Gráfico 2 2 Estructura Básica de MOODLE Fuente: Sebastián Troya

Como vemos en el gráfico el sitio MOODLE se compone de categorías. Las categorías pueden contener sub categorías pero están dentro del mismo grupo.

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

Una categoría tiene cursos. Un curso sólo puede estar en una categoría. Los cursos pueden estar distribuidos de forma semanal o por temas. Esto se lo hace para una mejor organización interna del curso.

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

Dependiendo de la distribución del curso ya sea “Semanal”, cada semana tiene sus actividades y recursos, o por “Temas”, cada tema tiene sus actividades y recursos.

MOODLE tiene diferentes niveles de acceso a los componentes de la plataforma. A estos niveles de acceso se les denomina “Rol”. MOODLE tiene definido 6 tipos de roles para los usuarios.

Administrador: El administrador del sistema tiene todos los permisos. Puede acceder a cualquier parte del sitio.

Creador de curso: 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.

Además MOODLE permite colocar permisos a los roles en seis contextos diferentes.

- Global
- Categorías.
- Cursos.
- Bloques y actividades.
- Usuario.
- Portada.

Es decir un rol puede tener permisos adicionales en los diferentes contextos.

Capacidades:

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

Para ver una lista completa de las capacidades revise la página oficial en esta dirección:

<http://docs.moodle.org/20/en/Category:Capabilities>.

Base de Datos.

MOODLE integra más de **doscientas tablas** en su base de datos incluyendo las necesarias por el núcleo y las de los módulos principales. La estructura de las tablas de los módulos se define en el archivo “instal.xml” dentro del directorio “db” en el directorio principal de los módulos. La base de datos puede ser editada y revisada a través del menú de administración de la plataforma.

La capa de base de datos en MOODLE ha sido escrita utilizando la librería de PHP ADOdb¹⁷ (<http://adodb.sourceforge.net/>), que fue creada para proporcionar un método estándar de acceso a varios sistemas de bases de datos, utilizando una interfaz de programación consistente. Las librerías de bases de datos nativas en PHP son específicas de una base de datos, por lo que es difícil usarlas para escribir un programa que soporte múltiples servidores de bases de datos. Gracias al uso de ADOdb, MOODLE proporciona soporte para varias bases de datos, entre las que se incluyen: MySQL, PostgreSQL, Microsoft SQL y Oracle.

Los requisitos de la versión de la base de datos para Moodle 1.9 y para Moodle 2.0 son:

- MySQL 4.1.16
- PostgreSQL 8.0
- Microsoft SQL 9.0

¹⁷ **ADOdb** es un conjunto de bibliotecas de bases de datos para PHP y Python. Esta permite a los programadores desarrollar aplicaciones web de una manera portable, rápida y fácil. La ventaja reside en que la base de datos puede cambiar sin necesidad de reescribir cada llamada a la base de datos realizada por la aplicación. (<http://es.wikipedia.org/wiki/ADOdb>)

- Oracle 9.0

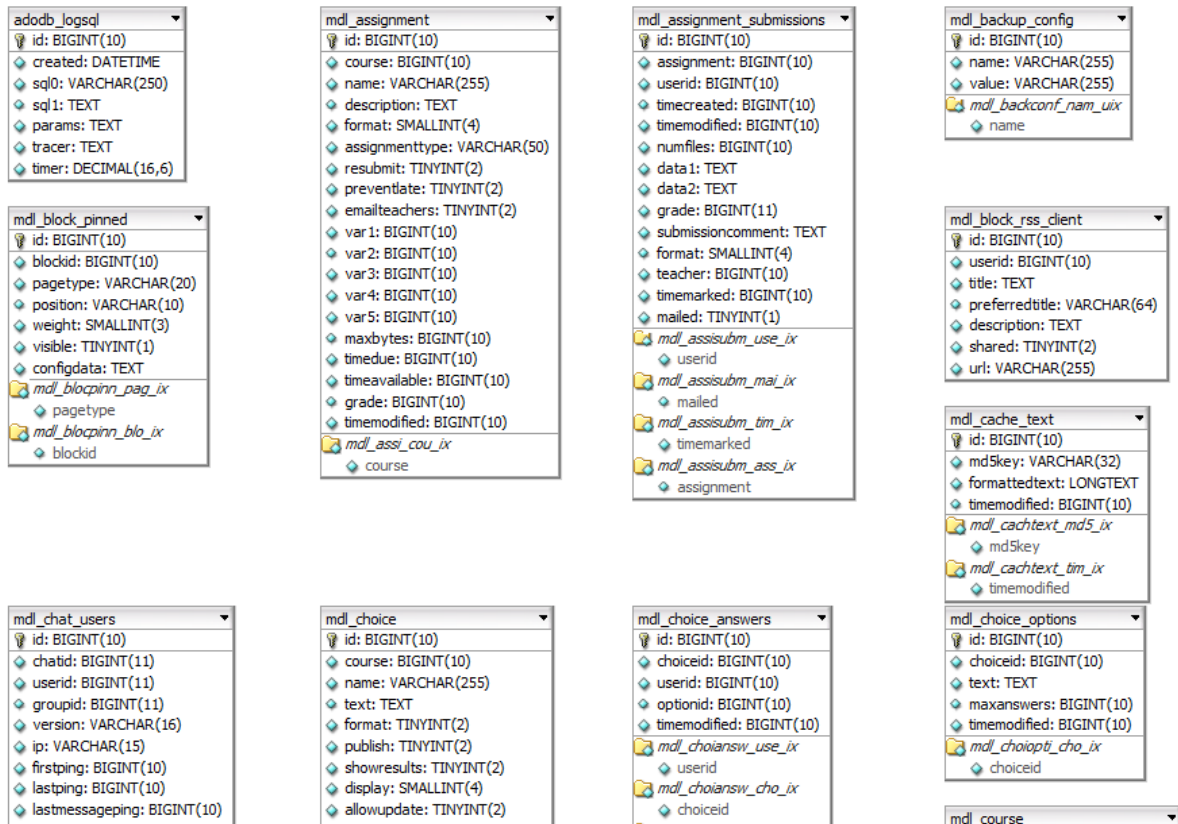


Gráfico 2 3 Modelo de base de datos Moodle 1.9 Fuente: Sebastián Troya

Como vemos en el gráfico no existe una relación física entre las tablas, pero obviamente existe una relación conceptual entre ellas. Como no se puede visualizar la relación entre tablas es un poco complicado entender cómo se relacionan las tablas pero con un poco de práctica y dedicación si se puede entender al modelo.

Capítulo III: Levantamiento de Requerimientos.

3.1 ¿Qué son los Requerimientos?

Los Requerimientos son una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Los requerimientos describen las acciones que el sistema realiza en las entradas para producir salidas deseadas.

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

Características de los Requerimientos.

Las características de un requerimiento son sus propiedades principales. A continuación se presentan las más importantes.

- **Necesario:** Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su utilidad, características físicas o calidad no pueden ser reemplazados por otras utilidades del producto o del proceso.
- **Conciso:** Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

- **Completo:** Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- **Consistente:** Un requerimiento es consistente si no es contradictorio con otro requerimiento.
- **No ambiguo:** Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.
- **Verificable:** Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

3.2 Métodos para levantar Requerimientos.

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema.

El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él.

A continuación se presentan las técnicas que han sido utilizadas para recolectar los requerimientos en el proceso de desarrollo de todo tipo de software.

Entrevista:

La Entrevista resulta una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada.

Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados.

A pesar de que las entrevistas son esenciales en el proceso de la captura de requisitos y con su aplicación el equipo de desarrollo puede obtener una amplia visión del trabajo y las necesidades del usuario, es necesario destacar que no es una técnica sencilla de aplicar.

Requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado. Aquí desempeña un papel fundamental la preparación de la entrevista.

JAD (Joint Application Development/Desarrollo conjunto de aplicaciones):

Esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes (IBM, 1997).

Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la aplicación de la técnica se trabajará sobre lo que se generará.

Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación. Durante la sesión se discute en grupo sobre estos temas, llegándose a una serie de conclusiones que se documentan.

En cada sesión se van concretando más las necesidades del sistema. Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales, ya que ahorra tiempo al

evitar que las opiniones de los clientes se tengan que contrastar por separado, pero requiere un grupo de participantes bien integrados y organizados.

Tormenta de Ideas:

La Tormenta de Ideas es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre.

Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador. Como técnica de captura de requisitos es sencilla de usar y de aplicar, contrariamente al JAD, puesto que no requiere tanto trabajo en grupo como éste.

Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros.

Concept Maps:

Los concept maps son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar.

Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje de éste.

Sin embargo, deben ser usados con cautela porque en algunos casos pueden ser muy sub-gestivos y pueden llegar a ser ambiguos en casos complejos, si no se acompaña de una descripción textual.

Caso de Uso:

Aunque inicialmente se desarrollaron como técnica para la definición de requisitos (Jacobson, 1995), algunos autores proponen casos de uso como técnica para la captura de requisitos.

Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema.

Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los *actores* del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores.

La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria si no se acompañan con una información textual o detallada con otra técnica como pueden ser los diagramas de actividades.

Cuestionarios y Checklists:

Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist).

Este cuestionario será completado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.

Comparación de Terminología:

Uno de los problemas que surge durante la recolección de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología.

Esta técnica es utilizada en forma complementaria a otras técnicas para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos (correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos (contraste).

3.3 Variables a dar seguimiento.

Una vez realizada una segunda entrevista con el personal del “Departamento de Nuevas Tecnologías” se han encontrado las variables que a las que necesitan dar seguimiento, el cual es el objetivo del presente trabajo.

Información General.

Esta variable contendrá de forma global el estado del sitio “PUCEMoodle”. Esta información es de suma importancia ya que podremos ver de forma consolidada el estado de la plataforma.

Estructura del Sitio.

Con esta variable podremos conocer como está organizado el sitio.

Estructura y tráfico del Curso.

Con esta variable podremos conocer como está organizado el curso. Además podremos ver como es el tráfico al curso.

3.4 Catálogo de Requerimientos.

3.4.1 Antecedentes.

En el momento actual los administradores de la plataforma “PUCEMoodle” no poseen herramientas para obtener información acerca del uso de la plataforma por parte de los usuarios de MOODLE.

MOODLE aunque tenga módulos donde se muestra información sobre las acciones de los usuarios en la plataforma, lo hace de manera muy general y sin filtros que se adapten a las necesidades de la PUCE.

MOODLE posee una tabla donde se almacena todas las acciones que el usuario realiza en la plataforma. Tomando los datos de esta tabla la plataforma tiene un módulo en el que se muestra un historial de forma muy general.

Por esta razón los administradores de “PUCEMoodle” no tienen las herramientas necesarias para obtener información que se adapten a las necesidades de estos.

El objetivo de este trabajo es la ayuda a los administradores de la plataforma para que tengan una herramienta con la que se pueda obtener la información necesaria para poder administrar de mejor manera la plataforma.

3.4.2 Estrategia.

Tomando en cuenta las variables a las que se van a dar seguimiento se ha decidido realizar el proyecto total en tres ciclos, para de esta forma dividir al proyecto en proyectos más pequeños para poder manejarlos de mejor manera.

En la primera iteración se realizarán todos los reportes que solventarán la variable “Información General”.

En la segunda iteración se realizarán todos los reportes que solventarán la variable “Estructura del Sitio”.

En la tercera iteración se realizarán todos los reportes que solventarán la variable “Estructura y tráfico del Curso”.

Cabe destacar que el modelo con el que se realizará el proyecto es “Iterativo Incremental” cada iteración tendrá su propio ciclo de vida, pudiendo revisar el ciclo anterior en el nuevo ciclo.

3.4.3 Requerimientos Funcionales

Los requerimientos funcionales establecen el comportamiento del sistema. Luego de varias reuniones se establecieron los siguientes requisitos funcionales.

- El módulo deberá ser utilizado sólo por el “Administrador”.
- El módulo debe ser capaz de generar reportes PDF.
- El modulo generará las siguientes consultas.
 1. Cursos activos e inactivos.
 2. Usuarios en cursos activos e inactivos.
 3. Número de Usuarios con roles de Profesor y Estudiante.
 4. Cursos más visitados.

5. Categorías de la plataforma con el número de cursos y el estado de los mismos (Activo o Inactivo).
 6. Recursos que tiene un curso.
 7. Tráfico de cada recurso.
 8. Estudiantes por curso.
 9. Creación de Cursos.
 10. Tráfico de los Cursos.
 11. Un Profesor a que cursos pertenece.
 12. Un Estudiante a que cursos pertenece.
- Las consultas se las puede hacer de forma general, en un periodo o comparación entre dos periodos.
 - Las consultas generarán gráficos estadísticos.
 - Los reportes en PDF tendrán cuadros estadísticos.

3.4.4 Requisitos No Funcionales.

Los requisitos no funcionales que se establecieron son los siguientes:

- El módulo podrá ser capaz de adaptarse si existen cambios en la configuración del servidor.
- El módulo se integrará dentro de la plataforma “PUCEMoodle”.

3.4.5 Requerimientos de Funcionamiento.

A continuación se listan los programas que se requieren para que el módulo funcione perfectamente.

- Adobe Flash Player 6 o superior.
- Version de Moodle 1.9 *.
- PHP 5.1.4 (Windows) y PHP 4.4.4 (Linux).
- Internet Explorer 8 o superior.

3.4.6 Prototipo.

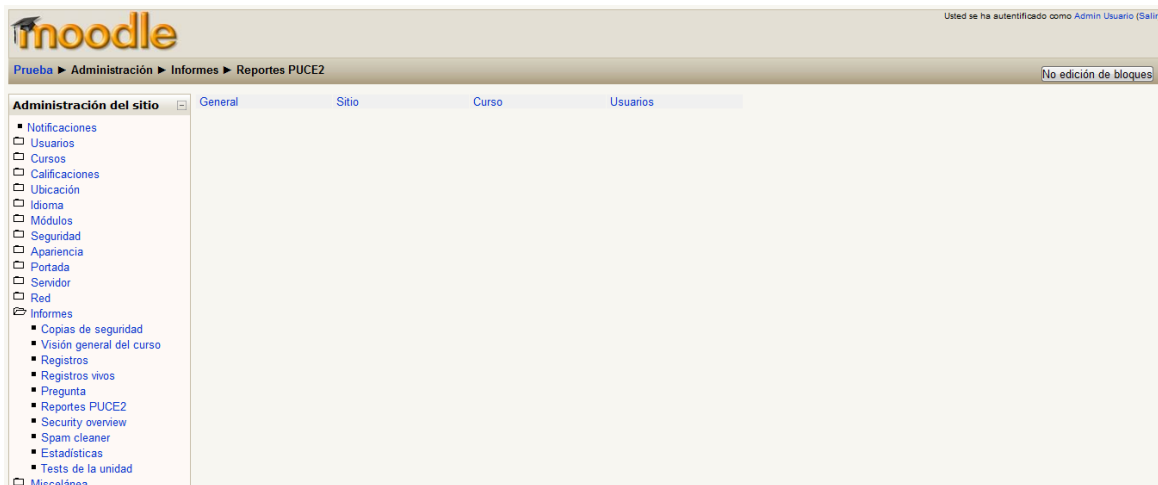


Gráfico 3 1 Prototipo Menú Principal Fuente: Sebastián Troya

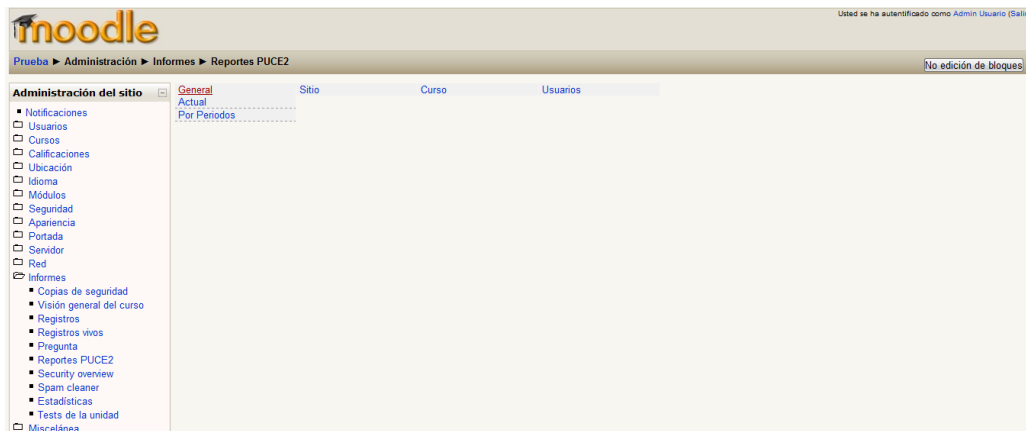


Gráfico 3 2 Prototipo General Fuente: Sebastián Troya

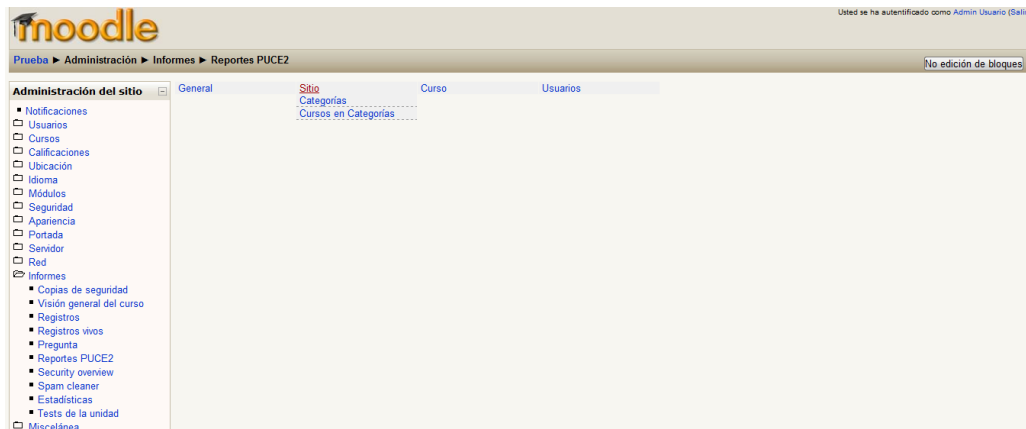


Gráfico 3 3 Prototipo Sitio Fuente: Sebastián Troya

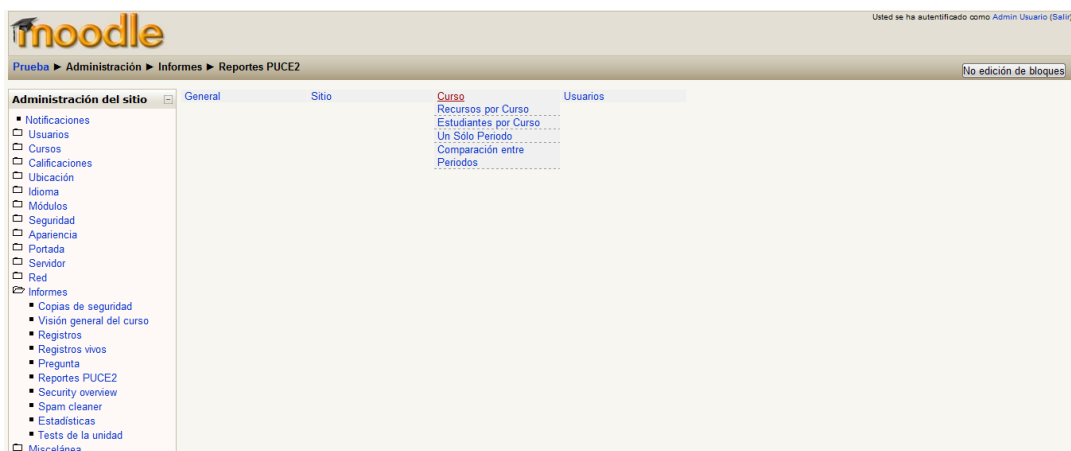


Gráfico 3 4 Prototipo Curso Fuente: Sebastián Troya

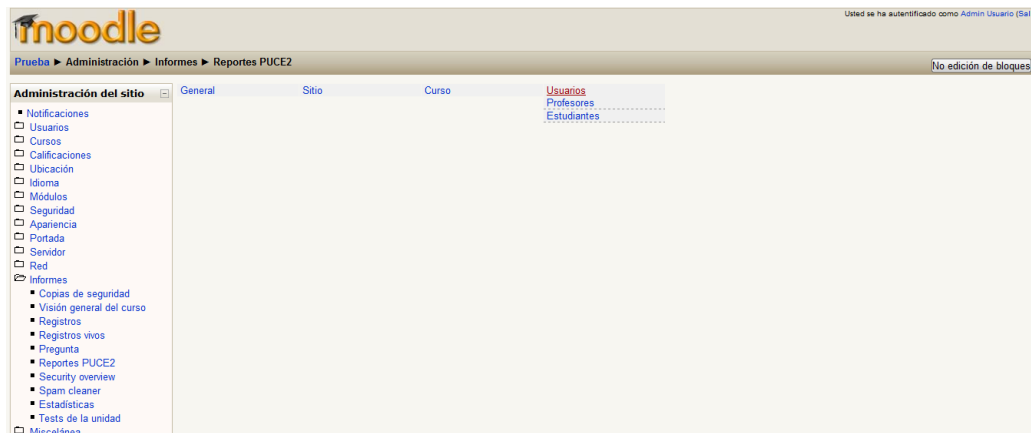


Gráfico 3 5 Prototipo Usuarios Fuente: Sebastián Troya

3.4.7 Diseño de Alto Nivel.

Diseño de Alto Nivel

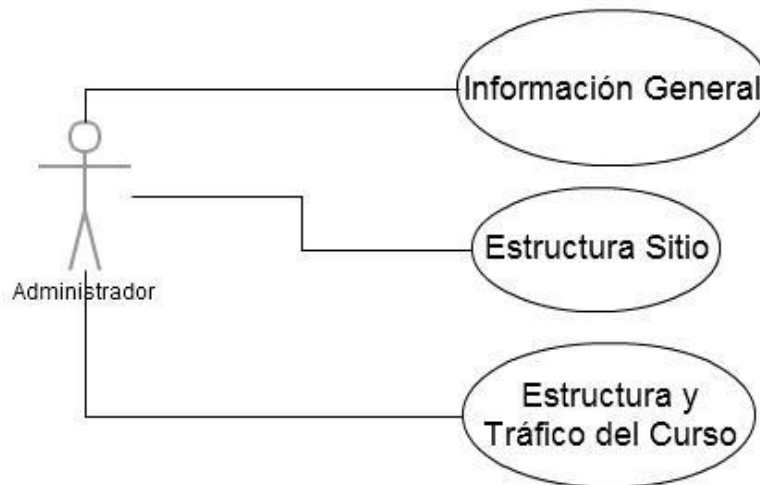


Gráfico 3 6 Diseño de Alto Nivel Fuente: Sebastián Troya

Capítulo IV: Desarrollo del Módulo.

4.1 Diagramas de Caso de Uso.

Se los utiliza para obtener información de los requisitos que el usuario necesita de forma general. Los diagramas de caso de uso son muy útiles para describir sistemas interactivos, ya que enseña lo que el usuario pretende obtener del sistema cuando él lo use.

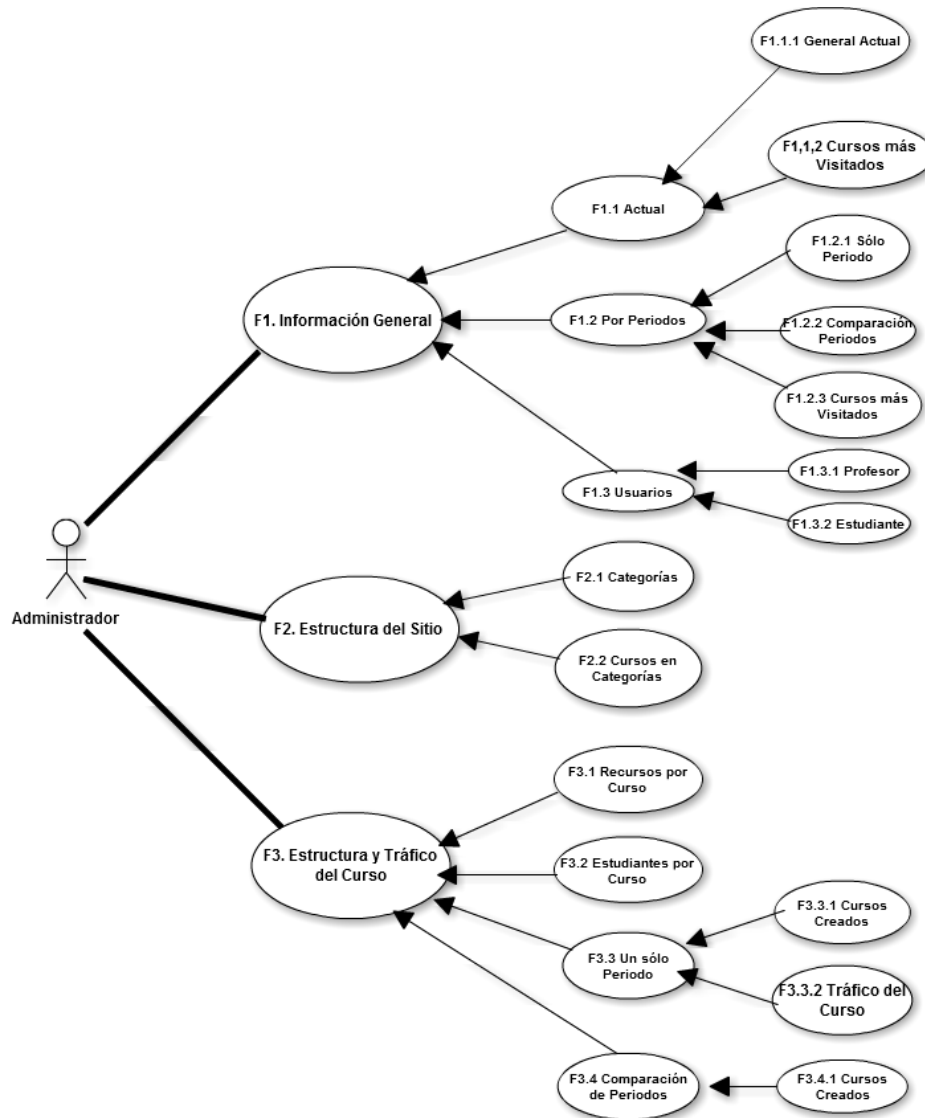


Gráfico 4. 1 Diagrama de Casos de Uso. Fuente: Sebastián Troya

4.2 Actor.

El actor es un rol que se le da a un usuario para que pueda interactuar con el sistema. Por la naturaleza del proyecto, existe un solo actor que se le denomina “Administrador”.

4.3 Desarrollo del Módulo.

4.3.1 Primer Ciclo.

4.3.1.1 Análisis.

Para el primer ciclo vamos a desarrollar las consultas de “Información General”. Dentro de esta consulta se encuentran las siguientes sub-consultas.

- Actual
 - General Actual.
 - Cursos más visitados.
- Por Periodos.
 - Un solo Periodo
 - Comparación entre periodos
 - Cursos más visitados
- Usuarios
 - Profesor
 - Estudiante.

4.3.1.2 Diseño.

Escenario F1.

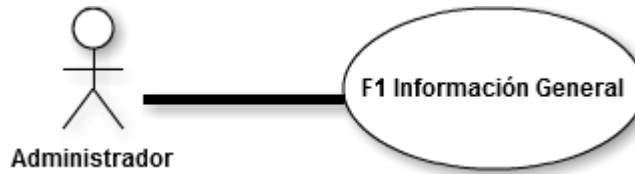


Gráfico 4. 2 Escenario F1. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Información General |
| Actor | Administrador |
| Precondición | Ninguno |
| Tipo | Primario |
| Detalle | El administrador puede hacer consultas acerca de la información general del sitio. |

Escenario F1.1.1

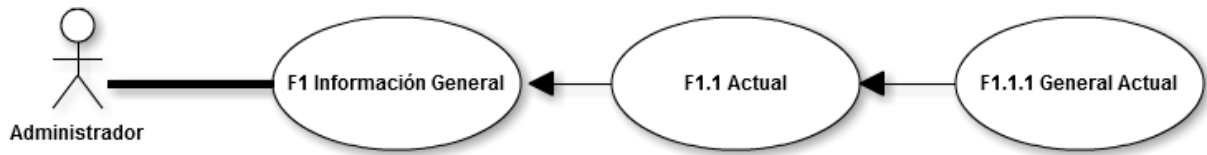


Gráfico 4. 3 Escenario F1.1.1. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | General Actual |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas acerca de la información general del sitio hasta la actualidad. |

Escenario F1.1.2

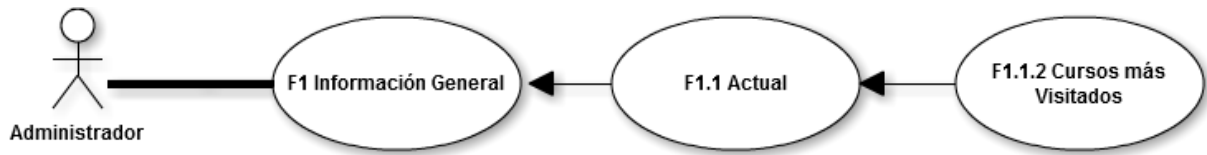


Gráfico 4. 4. Escenario F1.1.2. Fuente: Sebastián Troya

| | |
|--------------|---|
| Caso de Uso | Cursos más visitados |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas acerca de los cursos más visitados hasta la fecha de consulta. |

Escenario F1.2.1

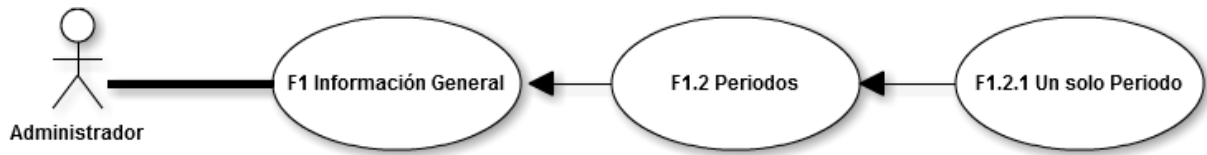


Gráfico 4. 5. Escenario F1.2.1. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Un solo Periodo |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas del estado general del sitio en un periodo en específico. |

Escenario F1.2.2



Gráfico 4. 6. Escenario F1.2.2. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Comparación entre periodos |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas para comparar entre dos periodos el estado general del sitio. |

Escenario F1.2.3

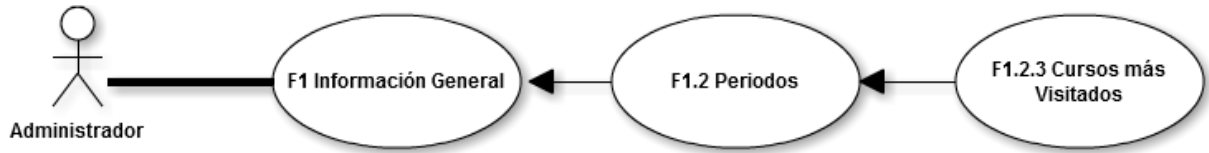


Gráfico 4. 7. Escenario F1.2.3. Fuente: Sebastián Troya

| | |
|--------------|---|
| Caso de Uso | Cursos más Visitados |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas de los cursos más generales en un periodo en específico. |

Escenario F1.3.1

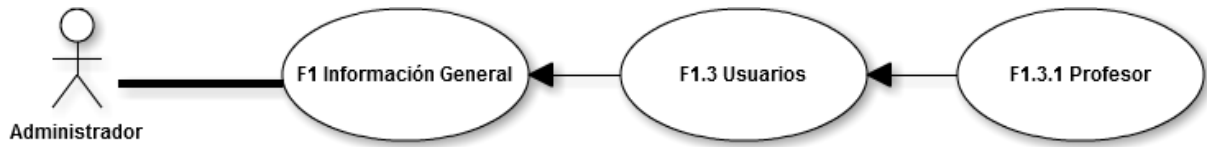


Gráfico 4. 8. Escenario F1.3.1. Fuente: Sebastián Troya

| | |
|--------------|---|
| Caso de Uso | Profesor |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas de los usuarios que tiene un rol de Profesor en que curso se encuentran. |

Escenario F1.3.2

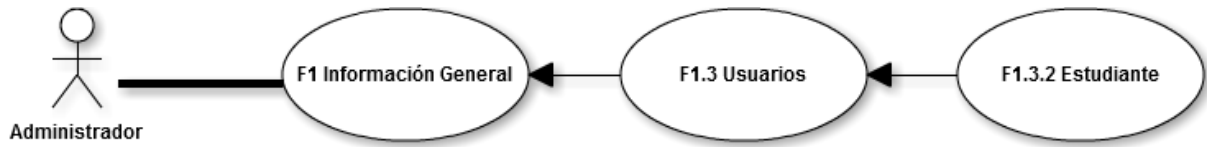


Gráfico 4. 9. Escenario F1.3.2. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Estudiante |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas de los usuarios que tiene un rol de Estudiante en que cursos se encuentran. |

4.3.1.3 Pruebas.

Se han realizado pruebas de aceptación sobre los casos de uso para verificar si los resultados obtenidos son los resultados esperados. A continuación se muestran los resultados.

| Caso de Uso | Nombre | Resultado Esperado | Resultado Obtenido | Aprobado |
|-------------|----------------------------|--|--|----------|
| F1.1.1 | General Actual | Información general de los Cursos, Usuarios, y Roles de la plataforma. | Información general de los Cursos, Usuarios, y Roles de la plataforma. | SI |
| F1.1.2 | Cursos más visitados | Lista de los 10 cursos más visitados hasta la fecha de la consulta. | Lista de los 10 cursos más visitados hasta la fecha de la consulta. | SI |
| F1.2.1 | Un solo Periodo | Información general de los Cursos, Usuarios, y Roles de la plataforma en un periodo establecido. | Información general de los Cursos, Usuarios, y Roles de la plataforma en un periodo establecido. | SI |
| F1.2.2 | Comparación entre periodos | Comparación del estado de los cursos entre dos periodos. | Comparación del estado de los cursos entre dos periodos. | SI |
| F1.2.3 | Cursos más Visitados | Comparación de los cursos más visitados en un periodo. | Comparación de los cursos más visitados en un periodo. | SI |

| | | | | |
|--------|------------|--|--|----|
| F1.3.1 | Profesor | Lista de los usuarios que tienen el rol de Profesor. | Lista de los usuarios que tienen el rol de Profesor. | SI |
| F1.3.2 | Estudiante | Lista de los usuarios que tienen el rol de Estudiante. | Lista de los usuarios que tienen el rol de Estudiante. | SI |

4.3.2 Segundo Ciclo.

4.3.2.1 Análisis.

Para el segundo ciclo vamos a desarrollar las consultas de “Estructura del Sitio”.

Dentro de esta consulta se encuentran las siguientes sub-consultas.

- Categorías.
- Cursos en Categorías.

4.3.2.2Diseño.

Escenario F2



Gráfico 4. 10. Escenario F2. Fuente: Sebastián Troya

| | |
|--------------|---|
| Caso de Uso | Estructura del Sitio |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Primario |
| Detalle | El administrador puede hacer consultas acerca de la estructura de organización del sitio. |

Escenario F2.1



Gráfico 4. 11. Escenario F2.1. Fuente: Sebastián Troya

| | |
|--------------|---|
| Caso de Uso | Categorías |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas de las categorías se encuentran en el sitio. |

Escenario F2.2



Gráfico 4. 12. Escenario F2.2. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Cursos en Categorías |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas de los cursos y en que categorías se encuentran los mismos. |

4.3.2.3 Pruebas.

Se han realizado pruebas de aceptación sobre los casos de uso para verificar si los resultados obtenidos son los resultados esperados. A continuación se muestran los resultados.

| Caso de Uso | Nombre | Resultado Esperado | Resultado Obtenido | Aprobado |
|-------------|----------------------|--|--|----------|
| F2.1 | Categorías | Lista de las categorías, con las sub-categorías, número de cursos y su estado. | Lista de las categorías, con las sub-categorías, número de cursos y su estado. | SI |
| F2.2 | Cursos en Categorías | Lista de los cursos que se encuentran en una categoría. | Lista de los cursos que se encuentran en una categoría. | SI |

4.3.3 Tercer Ciclo.

4.3.3.1 Análisis.

Para el tercer ciclo vamos a desarrollar las consultas de “Estructura y tráfico del Curso”. Dentro de esta consulta se encuentran las siguientes sub-consultas.

- Recursos por Curso.
- Estudiantes por Curso.
- Un solo Periodo.
 - Cursos Creados.
 - Tráfico del Curso.
- Comparación entre Periodos.
 - Cursos Creados.

4.3.3.2Diseño.

Escenario F3



Gráfico 4. 13. Escenario F3. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Estructura y tráfico del Curso |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Primario |
| Detalle | El administrador puede hacer consultas acerca de cómo están organizados los cursos y de cómo es el tráfico de estos. |

Escenario F3.1



Gráfico 4. 14. Escenario F3.1. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Recursos por Curso |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas de los recursos que tiene un curso. |

Escenario F3.2



Gráfico 4. 15. Escenario F3.2. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Estudiantes por Curso |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas de los usuarios que tiene el rol de estudiante en un curso en específico. |

Escenario F3.3.1



Gráfico 4. 16. Escenario F3.3.1. Fuente: Sebastián Troya

| | |
|--------------|---|
| Caso de Uso | Cursos Creados |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas de los cursos creados en un periodo en específico. |

Escenario F3.3.2



Gráfico 4. 17. Escenario F3.3.2. Fuente: Sebastián Troya

| | |
|--------------|--|
| Caso de Uso | Tráfico del Curso |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas el tráfico de un curso en un periodo en específico. |

Escenario F3.4.1

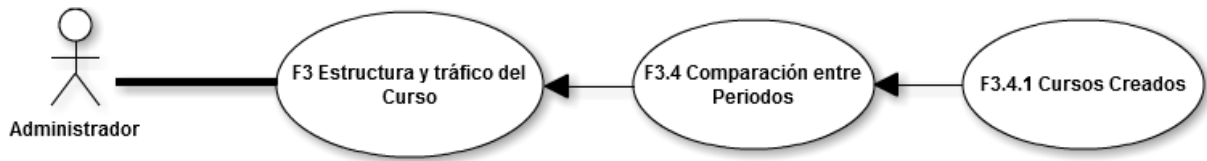


Gráfico 4. 18. Escenario F3.4.1. Fuente: Sebastián Troya

| | |
|--------------|---|
| Caso de Uso | Cursos Creados |
| Actor | Administrador |
| Precondición | Ninguna |
| Tipo | Secundario |
| Detalle | El administrador puede hacer consultas para comparar los cursos creados entre dos periodos en específico. |

4.3.3.3 Pruebas.

Se han realizado pruebas de aceptación sobre los casos de uso para verificar si los resultados obtenidos son los resultados esperados. A continuación se muestran los resultados.

| Caso de Uso | Nombre | Resultado Esperado | Resultado Obtenido | Aprobado |
|-------------|-----------------------|---|---|----------|
| F3.1 | Recursos por Curso | Número de cada recurso que tiene un curso en específico. | Número de cada recurso que tiene un curso en específico. | SI |
| F3.2 | Estudiantes por Curso | Lista de Usuarios con el rol de estudiante en un curso determinado. | Lista de Usuarios con el rol de estudiante en un curso determinado. | SI |
| F3.3.1 | Cursos Creados | Número de cursos creados en un periodo en específico. | Número de cursos creados en un periodo en específico. | SI |
| F3.3.2 | Tráfico del Curso | Tráfico de un curso en un periodo específico. | Tráfico de un curso en un periodo específico. | SI |
| F3.4.1 | Cursos Creados | Comparación de los cursos creados entre dos periodos. | Comparación de los cursos creados entre dos periodos. | SI |

4.4 Diagrama Entidad Relación.

Como ya se expuso, MOODLE tiene más de 200 tablas en su base de datos. La mayoría de estas tablas tiene relación con los módulos que se instalan en la plataforma. Pero para el proyecto que se está realizando, se identificó el siguiente modelo entidad relación.

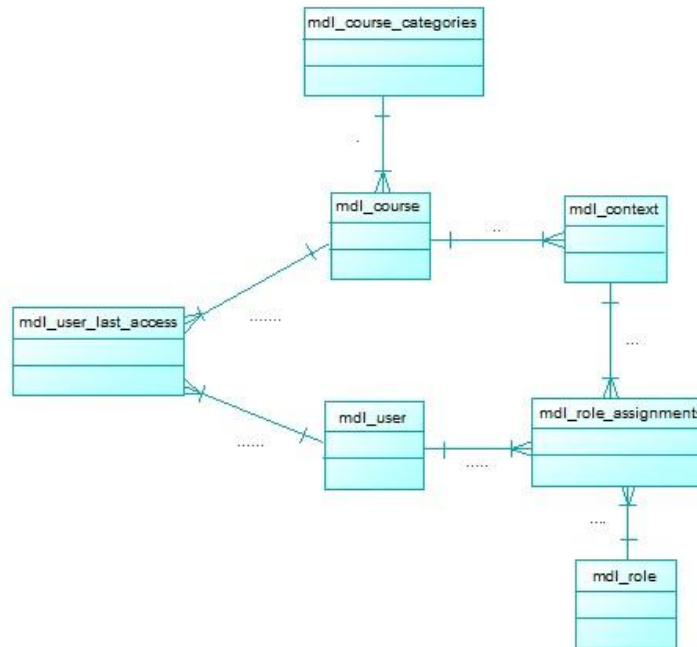


Gráfico 4. 19. Diagrama Entidad-Relación. Fuente: Sebastián Troya

Capítulo V: Conclusiones y Recomendaciones.

5.1 Conclusiones.

Una vez finalizado el desarrollo del módulo se seguimiento del uso de la plataforma MOODLE por parte de los docentes de la PUCE sede Quito, se procedió a la implementación del mismo.

Al implementar el módulo en la PUCE se ha logrado observar y analizar la información que existe en la herramienta MOODLE de la PUCE.

Con el desarrollo del módulo se cubren los siguientes puntos:

1. Seguimiento de los docentes dentro de la plataforma. Esto implica los cursos en los que se encuentran enrolados, número de recursos que utiliza en cada curso.
2. Generar cuadros estadísticos de cómo es el uso de la plataforma MOODLE con lo que facilitará la toma de decisiones por parte de la unidad a cargo.
3. Con este módulo podemos obtener informes de cuantos usuarios existen, sus roles, en que curso se encuentran enrolados.
4. El módulo permite visualizar el número de cursos existentes, el detalle de los estudiantes que pertenecen a un curso determinado.
5. Además podemos tener información verdadera del estado de la plataforma, ya que MOODLE, a pesar de tener un módulo de reportes no cumple con las necesidades de la PUCE.
6. Este proyecto generó el conocimiento necesario para la elaboración de módulos personalizados en MOODLE. Gracias a este conocimiento se

pueden desarrollar otro tipo de módulos que se adapten a las necesidades de la PUCE.

7. El desarrollo de este proyecto generó documentación técnica indispensable para el entendimiento de MOODLE, y con esto generar productos que se adapten a esta plataforma.

5.2 Recomendaciones.

Al implementar el módulo de seguimiento del uso de la plataforma por parte de los docentes de la PUCE y la generación de reportes, se sugiere tomar en consideración las siguientes recomendaciones.

En un principio el ingreso de los datos en la Plataforma MOODLE no se hizo cumpliendo con los estándares de la Dirección de Informática de la PUCE, esto ocasiona problemas en la generación de reportes.

Al existir diferentes maneras de escribir la información, en unos casos nombre en minúsculas, letra capital, mayúsculas, sin tildes, con abreviaturas, signos especiales, los reportes no se muestran de manera unificada. Se sugiere que se depure y se cambie estos datos según el estándar actual.

Nombre de cursos como:

“OP:CULTURA Y PODER (2011-02)”, “ADM. FINANC. DE PROYECTOS (2009-02)” en donde el nombre se forma con más de dos palabras que especifican el destino de estos, deberían tener un mismo formato. Se sugiere establecer que el signo de separación sea el mismo en todos los casos.

Los datos de los estudiantes y profesores son importados desde el sistema académico lo que permite mantener el estándar para los nombres y apellidos, Se recomienda que si dada la eventualidad se tiene que crear un usuario fuera de este proceso mantenga el estándar que se maneja.

Con los reportes que genera el módulo “SRMP” podemos observar que existen usuarios que se encuentran duplicados, otros que no se encuentran enrolados en cursos que se encuentran en estado inactivo. Se recomienda una depuración de la base de datos de los usuarios, y de los archivos que ellos hayan subido.

Cuando se termina un periodo académico los cursos se ponen en estado inactivo., esto en función de poder habilitarlos de nuevo en próximos semestres, según la indicación del docente. Hay casos en los que los cursos están inactivos por varios semestres y no hay una petición para habilitarlos, por lo que los recursos que se utilizan para estos cursos ocupen espacio innecesario. Se recomienda tener una política para este tipo de casos y de esa manera poder borrar esos cursos y liberar el espacio en el servidor.

Las ventajas que posee MOODLE es la posibilidad de desarrollar módulos personalizados, según el modo de trabajo en cada una de las unidades académicas.

Anexos

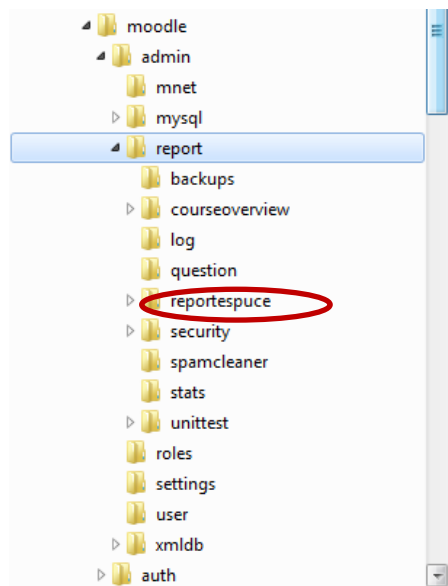
Manual de Usuario.

Introducción.

El presente manual está destinado a los administradores de la plataforma MOODLE para el uso del módulo de Seguimiento.

Instalación.

Para instalar el módulo copie todos los archivos en la carpeta “admin/report”.



Manual de Usuario Gráfico 1 Donde Instalar el Módulo Fuente: Sebastián Troya

Este es el único requisito que se necesita para instalar el módulo. No se necesita realizar ninguna configuración adicional ya que todos los parámetros se configuran automáticamente.

Requerimientos de Funcionamiento.

A continuación se listan los programas que se requieren para que el módulo funcione perfectamente.

- Adobe Flash Player 6 o superior.
- Version de Moodle 1.9 *.
- PHP 5.1.4 (Windows) y PHP 4.4.4 (Linux).
- Internet Explorer 8 o superior.

Funcionamiento.

El presente módulo cuenta con 12 aplicaciones divididas en 4 categorías que para mayor comodidad se las puso en el menú.



Manual de Usuario Gráfico 2 Menú Fuente: Sebastián Troya

General.

Dentro del menú “General” tenemos reportes de cómo es el estado de la plataforma tanto en el estado actual como en un periodo específico y para comparar entre periodos.



Manual de Usuario Gráfico 3 Menú General Fuente: Sebastián Troya

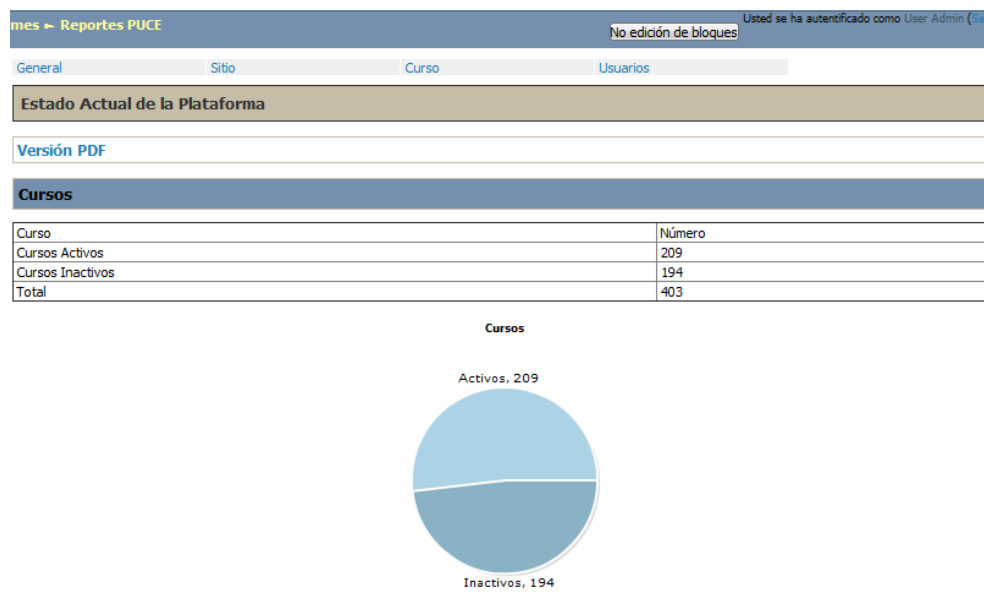
Dentro del Sub Menú Actual tenemos los reportes de “General Actual” y “Cursos más Visitados”

| | | | |
|------------------------------|--------------------------------------|-------|----------|
| General | Sitio | Curso | Usuarios |
| Actual | General Actual | | |
| Por Periodos | Cursos más visitados | | |

Manual de Usuario Gráfico 4Submenú Actual Fuente: Sebastián Troya

General Actual

En este reporte se muestran cual es el estado actual de la plataforma tomado en cuenta las variables “Cursos”, “Usuarios”, “Roles”.

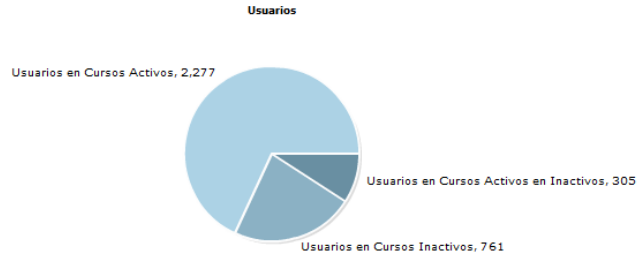


Manual de Usuario Gráfico 5 General Actual Fuente: Sebastián Troya

En la parte de Usuarios hay la opción de ver a los Usuarios que están dentro de la plataforma pero no están asignados a ningún curso. Para ver la lista de estos usuarios basta con hacer [click](#) en el hipervínculo.

| Usuarios | |
|--|--------|
| Usuario | Número |
| Usuarios en Cursos Activos | 2277 |
| Usuarios en Cursos Inactivos | 761 |
| Usuarios en Cursos Activos e Inactivos | 305 |
| Usuarios sin Curso | 2081 |
| Total Usuario | 5424 |

Usuarios sin Curso

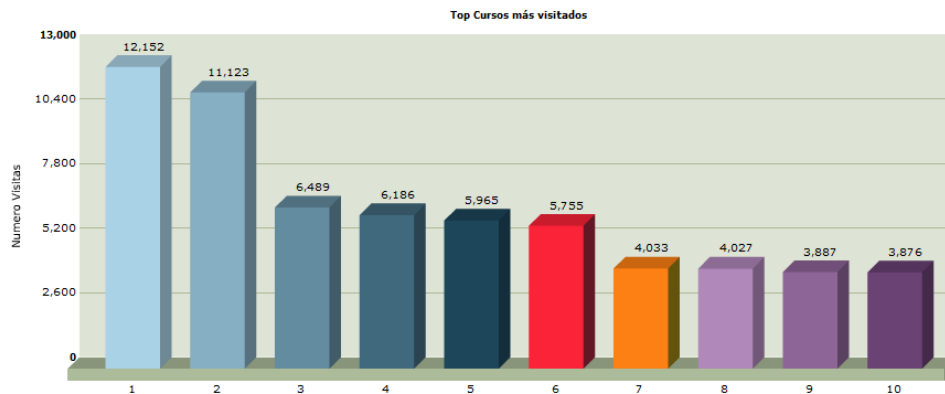


Manual de Usuario Gráfico 6 Usuarios Sin Curso Fuente: Sebastián Troya

Cursos más Visitados.

En este reporte se muestra un top de los 10 cursos más visitados desde el momento en que se instaló MOODLE hasta el día en que se hace la consulta.

| Cursos más Visitados | | | |
|---------------------------------|---|----------------------------------|-------------------|
| Versión PDF | | | |
| TOP de los Cursos más visitados | | | |
| Nº | Curso | Profesor | Número de Visitas |
| 1 | INSTALACIONES ELÉCTRICAS (2011-01) | BENIGNO SANTOS | 12152 |
| 2 | INGLÉS 1 | OVANDO RODRÍGUEZ MARLON LEONARDO | 11123 |
| 3 | ELECT. II: MULTIMEDIA PARA ADMINIST (2011-01) | YOLANDA CECILIA GARCIA PAREDES | 6489 |
| 4 | INGLÉS 2 | OVANDO RODRÍGUEZ MARLON LEONARDO | 6186 |
| 5 | EL. II: INNOVACIÓN Y CREATIVIDAD (PROFESOR: E. HERNANDEZ) (2011-01) | LUIS ENRIQUE HERNANDEZ DIAZ | 5965 |
| 6 | EDUCACIÓN MEDIADA POR COMPUTADOR | LUIS IVÁN CARRERA BARRIONUEVO | 5755 |
| 7 | FUNDAMENTOS BIOLÓGICOS DE LA PSICOLOGÍA (2) (2011-01) | MARÍA ALEJANDRA CAMACHO MONGE | 4033 |
| 8 | ADMINIST.DE PROCESOS 9-11 (2011-01) | PAUL MICHAEL IDROBO DÁVALOS | 4027 |
| 9 | FÍSICA MATEMÁTICA I (3) (2011-01) | ELVIA PATRICIA PROAÑO ROSADO | 3887 |
| 10 | FISICOQUÍMICA II (2011-01) | LORENA MENESES OLMEDO | 3876 |



Manual de Usuario Gráfico 7 Cursos Más Visitados Fuente: Sebastián Troya

Por Periodos.

En el Sub-menú “Por Periodos” tenemos “Un solo Periodo”, “Comparación entre Periodos” y “Cursos más visitados en un Periodo”.



Manual de Usuario Gráfico 8 Sub-Menú Por Periodos Fuente: Sebastián Troya

Un solo Periodo.

Este reporte sirve para ver el estado de la plataforma en un periodo en específico.



Manual de Usuario Gráfico 9 Un solo Periodo Ingreso Datos Fuente: Sebastián Troya

Se llenan los parámetros para empezar a realizar la búsqueda.

- Desde: Parámetro para escoger la fecha de Inicio para la búsqueda.
- Hasta: Parámetro para escoger la Fecha de Fin para la búsqueda.
- Nombre: Nombre del periodo en que se hace la búsqueda.

Para escoger las fechas, basta con hacer clic sobre el ícono del calendario que se encuentra a la derecha de cada parámetro. Las fechas tienen el formato Mes/Día/Año.

Una vez escogidos los parámetros se hace clic en el botón “Aceptar” y se procederá a realizar la búsqueda entre las fechas en las que se ha escogido.

Comparación entre Periodos.

Este reporte sirve para comparar el estado de la plataforma que muestra la cantidad de usuarios, cursos, etc. entre dos periodos.

The screenshot shows a web form titled "Estado de la Plataforma entre Periodos". It contains two sections for period selection. The first section, "Periodo 1", has three input fields: "Desde:" with the value "01/01/2010", "Hasta:" with the value "12/31/2010", and "Nombre:" with the value "AÑO 2010". The second section, "Periodo 2", also has three input fields: "Desde:" with the value "01/01/2010", "Hasta:" with the value "12/31/2010", and "Nombre:" with the value "AÑO 2011". Below these sections is a button labeled "Aceptar".

Manual de Usuario Gráfico 10 Comparación Entre Periodos General Fuente: Sebastián Troya

Periodo 1:

- Desde: Parámetro para escoger la fecha de Inicio del Periodo 1 para la búsqueda.
- Hasta: Parámetro para escoger la Fecha de Fin del Periodo 1 para la búsqueda.
- Nombre: Nombre del Periodo 1 en que se hace la búsqueda.

Periodo 2:

- Desde: Parámetro para escoger la fecha de Inicio del Periodo 2 para la búsqueda.
- Hasta: Parámetro para escoger la Fecha de Fin del Periodo 2 para la búsqueda.
- Nombre: Nombre del Periodo 2 en que se hace la búsqueda.

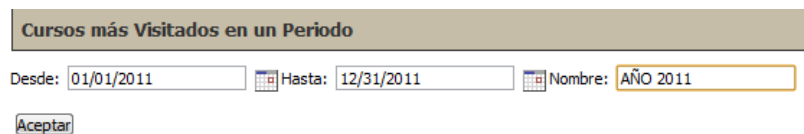
PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Para escoger las fechas, basta con hacer clic sobre el ícono del calendario que se encuentra a la derecha de cada parámetro. Las fechas tienen el formato Mes/Día/Año.

Una vez escogidos los parámetros se hace clic en el botón “Aceptar” y se procederá a realizar la búsqueda entre las fechas en las que se ha escogido.

Cursos más Visitados en un Periodo.

Este reporte sirve para ver el los 10 cursos más visitados en un periodo en específico.



The screenshot shows a web form titled "Cursos más Visitados en un Periodo". It contains three input fields: "Desde:" with the value "01/01/2011", "Hasta:" with the value "12/31/2011", and "Nombre:" with the value "AÑO 2011". Each field has a small calendar icon to its right. Below the fields is a button labeled "Aceptar".

Manual de Usuario Gráfico 11 Cursos más Visitados en un Periodo Fuente: Sebastián Troya

Se escogen los parámetros para empezar a realizar la búsqueda.

- Desde: Parámetro para escoger la fecha de Inicio para la búsqueda.
- Hasta: Parámetro para escoger la Fecha de Fin para la búsqueda.
- Nombre: Nombre del periodo en que se hace la búsqueda.

Para escoger las fechas, basta con hacer clic sobre el ícono del calendario pegar ícono del calendario que se encuentra a la derecha de cada parámetro. Las fechas tienen el formato Mes/Día/Año.

Una vez escogidos los parámetros se hace clic en el botón “Aceptar” y se procederá a realizar la búsqueda entre las fechas en las que se ha escogido.

Sitio.

Dentro del menú “Sitio” tenemos reportes de cómo es la distribución de los cursos dentro de la plataforma. Dividido en “Categorías” y “Cursos en Categorías”.

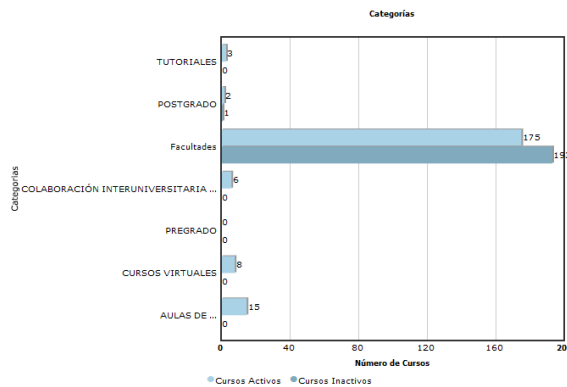


Manual de Usuario Gráfico 12 Menú Sitio Fuente: Sebastián Troya

Categorías.

Este reporte sirve para ver la distribución de la plataforma en categorías, cuantas sub-categorías tienen, cuantos cursos están en cada categoría, y en qué estado se encuentran (Activo o Inactivo).

| Categorías | | | | |
|---|----------------|----------------|------------------|-----------------|
| Categoría | Sub-Categorías | Cursos Activos | Cursos Inactivos | Total de Cursos |
| TUTORIALES | 2 | 3 | 0 | 3 |
| POSTGRADO | 2 | 2 | 1 | 3 |
| FACULTADES | 21 | 175 | 193 | 368 |
| COLABORACIÓN INTERUNIVERSITARIA VIRTUAL | 0 | 6 | 0 | 6 |
| PREGRADO | 0 | 0 | 0 | 0 |
| CURSOS VIRTUALES | 0 | 8 | 0 | 8 |
| AULAS DE PRÁCTICA | 0 | 15 | 0 | 15 |
| Total | 25 | 209 | 194 | 403 |



Manual de Usuario Gráfico 13 Categorías Fuente: Sebastián Troya

Cursos en Categorías.

Este reporte nos permite ver cuáles son los cursos que se encuentran en cada sub-categoría, el estado, la fecha de Creación, Última Fecha de Ingreso, y la Última Fecha de Modificación del Curso.

Categorías Actuales de la Plataforma

Categoría:

Aceptar **Versión** **Cursos**

- TUTORIALES
- POSTGRADO
- FACULTADES
- COLABORACIÓN INTERUNIVERSITARIA VIRTUAL
- PREGRADO
- CURSOS VIRTUALES
- AULAS DE PRÁCTICA

| Curso | Estado | Subcategoría | Fecha de Creación | Última Fecha de Ingreso | Última Fecha de Modificación |
|--|----------|----------------|-------------------|-------------------------|------------------------------|
| FUNDAMENTOS DE MATEMÁTICA | Activo | JURISPRUDENCIA | 2010-07-30 | 2011-09-13 | 2011-01-18 |
| CONFLICT PREV. AND PEACEBUILDING (2010-01) | Activo | JURISPRUDENCIA | 2010-08-17 | 2011-12-02 | 2011-11-25 |
| GRUPOS SOCIALES CON DERECHOS ESPECÍFICOS (2011-01) | Activo | JURISPRUDENCIA | 2011-01-07 | 2011-09-22 | 2011-08-18 |
| DEONTOLOGÍA JURÍDICA (2010-02) | Inactivo | JURISPRUDENCIA | 2011-01-10 | 2011-08-03 | 2011-06-17 |
| INTRODUCCIÓN A LA PRÁCTICA DEL DERECHO (2011-01) | Activo | JURISPRUDENCIA | 2011-01-10 | 2011-08-10 | 2011-08-10 |
| METODOLOGÍA DE LA INVESTIGACIÓN JURÍDICA (2010-02) | Inactivo | JURISPRUDENCIA | 2011-01-10 | 2011-08-03 | 2011-06-17 |

Manual de Usuario Gráfico 14 Cursos en Categorías Fuente: Sebastián Troya

Para ver los cursos, se selecciona la categoría y se da clic en el botón “Aceptar”.

Curso.

Dentro del menú “Curso” tenemos reportes de “Recursos por Curso”, “Estudiantes por Curso”, “Curso creados en un Periodo”, “Tráfico del Curso en un Periodo”, y “Comparación de Cursos Creados en dos periodos”.

Recursos por Curso.

Con este reporte se pueden de forma visual ver el número de cada tipo de recursos que se están utilizando en el curso.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Recursos del Curso ANATOMIA Y FISILOGIA I (2011-01)

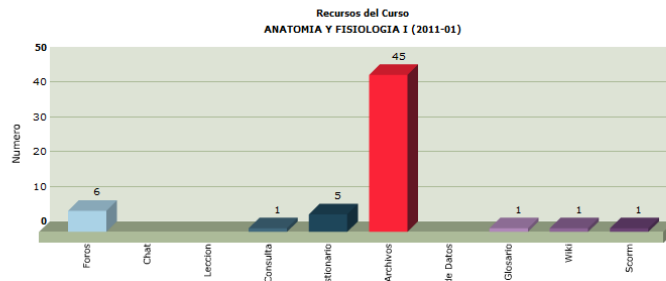
Profesor:
ELSIE PAULINA MOREANO BARRAGAN

Fecha de creación: 2011-06-15

Formato: TOPICS

Número de Secciones: 4

| Recurso | Total |
|---------------|-------|
| Foros | 6 |
| Chat | 0 |
| Lección | 0 |
| Consulta | 1 |
| Cuestionario | 5 |
| Archivos | 45 |
| Base de Datos | 0 |
| Glosario | 1 |
| Wiki | 1 |
| Scorm | 1 |



Manual de Usuario Gráfico 15 Recursos por Curso Fuente: Sebastián Troya

Para ver todos los recursos de cada tipo de recursos del Curso se hace clic sobre el tipo de recurso.

Estudiantes por Curso.

Con este reporte se puede ver los usuarios que se encuentran en el curso con el rol de “Estudiante”.

Estudiantes del Curso ALEMÁN 1 (2011-01)

Profesor:
SYLVIA LEHMANN PRUIS

Fecha de creación: 2011-09-23

Formato: WEEKS

Número de Secciones: 18

Total de Alumnos: 12

| Usuario | Alumno | Email | Último acceso |
|---------------|---|---------------------------|---------------------|
| MDAVILA764 | MARÍA FERNANDA DÁVILA LARA | mdavila764@PUCE.EDU.EC | 2011-11-26 09:11:17 |
| LMERAS685 | LUIS SEBASTIÁN MERA ZURITA | lmeras685@PUCE.EDU.EC | NUNCA |
| LFLORES349 | LUIS KEVIN ABSGAL FLORES SOLIS | lflores349@PUCE.EDU.EC | 2011-12-02 05:12:29 |
| LCUEVA552 | LEONEL EDUARDO CUEVA GUERRERO | lcueva552@PUCE.EDU.EC | NUNCA |
| JREYES395 | JUAN PABLO REYES OCHOA | jreyes395@PUCE.EDU.EC | NUNCA |
| JLARCO367 | JORDY ALEXANDER LARCO LASSO | jlarco367@PUCE.EDU.EC | 2011-12-01 11:12:56 |
| GRIVERA551 | GALO ENRIQUE RIVERA PAZMIÑO | grivera551@PUCE.EDU.EC | 2011-12-01 08:12:01 |
| ESALAZAR | ESTEFANÍA SALAZAR ROSERO | esalazar@PUCE.EDU.EC | NUNCA |
| ERODRIGUEZ001 | ESMERALDA ZARZANA RODRIGUEZ MAH-BORCIDA | erodriguez001@PUCE.EDU.EC | NUNCA |
| EDHEREDIA | EMILY DAYAN HEREDIA TUPIZA | edheredia@PUCE.EDU.EC | NUNCA |
| CRESTREPO089 | CAMILA RESTREPO NÉNEZ | crestrepo089@PUCE.EDU.EC | NUNCA |
| AVIERA001 | ANDREA DE LOS ANGELES VIERA PALMA | aviera001@PUCE.EDU.EC | NUNCA |

Manual de Usuario Gráfico 16 Estudiantes por Curso Fuente: Sebastián Troya

Un solo Periodo.

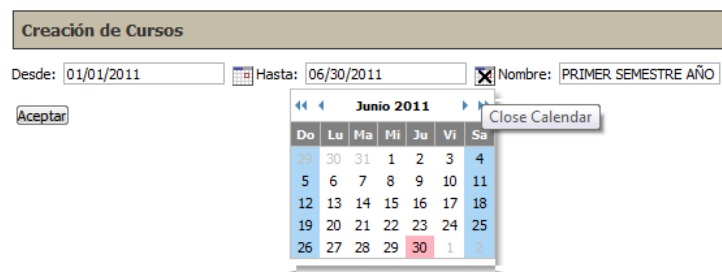
En el Sub-Menú “Un solo Periodo” tenemos los reportes de “Cursos Creados” y “Tráfico de Curso”.



Manual de Usuario Gráfico 17 Sub-Menú Un solo Periodo Fuente: Sebastián Troya

Cursos Creados.

En este reporte se visualizan el número de cursos que se han creado en un periodo en específico.



Manual de Usuario Gráfico 18 Cursos Creados Un solo Periodo Parámetros Fuente: Sebastián Troya

Se escogen los parámetros para empezar a realizar la búsqueda.

- Desde: Parámetro para escoger la fecha de Inicio para la búsqueda.
- Hasta: Parámetro para escoger la Fecha de Fin para la búsqueda.
- Nombre: Nombre del periodo en que se hace la búsqueda.

Para escoger las fechas, basta con hacer clic sobre el ícono del calendario que se encuentra a la derecha de cada parámetro. Las fechas tienen el formato Mes/Día/Año.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Una vez escogidos los parámetros se hace clic en el botón “Aceptar” y se procederá a realizar la búsqueda entre las fechas en las que se ha escogido.

Creación de Cursos

Desde: 01/01/2011 Hasta: 06/30/2011 Nombre: PRIMER SEMESTRE AÑO

Aceptar

Versión PDF

Cursos creados en el periodo PRIMER SEMESTRE AÑO 2011

| Curso | Fecha de Creación |
|---|-------------------|
| PRÁCTICAS PREPROFESIONALES II (2010-02) | 2011-01-07 |
| GRUPOS SOCIALES CON DERECHOS ESPECÍFICOS (2011-01) | 2011-01-07 |
| GENERAC. DE INGRESOS (2010-02) | 2011-01-07 |
| HIST. ARTE I (2010-02) | 2011-01-07 |
| LAS VANGUARDIAS ARTÍSTICAS DEL SIGLO XX (2010-02) | 2011-01-07 |
| FÍSICA II PARA CIENCIAS BIOLÓGICAS (2010-02) | 2011-01-10 |
| FÍSICA II: TEORÍA Y LABORATORIO, QUÍMICA ANALÍTICA. (2010-02) | 2011-01-10 |
| HISTORIA ARO. I (2010-02) | 2011-01-10 |

Manual de Usuario Gráfico 19 Cursos Creados en un solo Periodo Fuente: Sebastián Troya

Al hacer clic sobre el curso se pueden ver los recursos de este curso.

Tráfico del Curso.

En este reporte se puede ver el número de accesos al curso en un determinado periodo.

Tráfico de Cursos

Desde: 01/01/2011 Hasta: 12/31/2011 Nombre: AÑO 2011

Curso: ALEMÁN 1 (2011-01)

Aceptar

Manual de Usuario Gráfico 20 Tráfico del Curso en Un solo Periodo Fuente: Sebastián Troya

Se escogen los parámetros para empezar a realizar la búsqueda.

- Desde: Parámetro para escoger la fecha de Inicio para la búsqueda.
- Hasta: Parámetro para escoger la Fecha de Fin para la búsqueda.
- Nombre: Nombre del periodo en que se hace la búsqueda.
- Curso: Curso del que se quiere ver el reporte.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Para escoger las fechas, basta con hacer clic sobre el ícono del calendario que se encuentra a la derecha de cada parámetro. Las fechas tienen el formato Mes/Día/Año.

Una vez escogidos los parámetros se hace clic en el botón “Aceptar” y se procederá a realizar la búsqueda entre las fechas en las que se ha escogido.

Comparación entre Periodos.

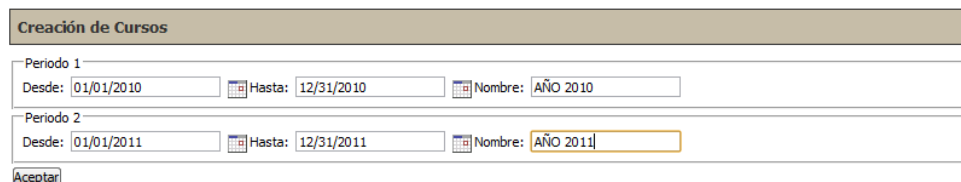
En el Sub-Menú “Comparación entre Periodos” tenemos el reporte “Creación de Cursos”.



Manual de Usuario Gráfico 21 Comparación entre Periodos Fuente: Sebastián Troya

Cursos Creados.

En este reporte se visualiza una comparación la creación de cursos entre dos periodos definidos.

A screenshot of a web form titled 'Creación de Cursos'. The form is divided into two sections: 'Periodo 1' and 'Periodo 2'. Each section contains three input fields: 'Desde' (start date), 'Hasta' (end date), and 'Nombre' (course name). For Periodo 1, the 'Desde' field is '01/01/2010', 'Hasta' is '12/31/2010', and 'Nombre' is 'AÑO 2010'. For Periodo 2, the 'Desde' field is '01/01/2011', 'Hasta' is '12/31/2011', and 'Nombre' is 'AÑO 2011'. Below the form is an 'Aceptar' button.

Manual de Usuario Gráfico 22 Creación de Cursos Comparación entre Periodos Fuente: Sebastián Troya

Periodo 1:

- Desde: Parámetro para escoger la fecha de Inicio del Periodo 1 para la búsqueda.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

- Hasta: Parámetro para escoger la Fecha de Fin del Periodo 1 para la búsqueda.
- Nombre: Nombre del Periodo 1 en que se hace la búsqueda.

Periodo 2:

- Desde: Parámetro para escoger la fecha de Inicio del Periodo 2 para la búsqueda.
- Hasta: Parámetro para escoger la Fecha de Fin del Periodo 2 para la búsqueda.
- Nombre: Nombre del Periodo 2 en que se hace la búsqueda.

Para escoger las fechas, basta con hacer clic sobre el ícono del calendario que se encuentra a la derecha de cada parámetro. Las fechas tienen el formato Mes/Día/Año.

Una vez escogidos los parámetros se hace clic en el botón “Aceptar” y se procederá a realizar la búsqueda entre las fechas en las que se ha escogido.

Usuarios.

Dentro del menú “Usuarios” tenemos reportes de “Profesores” y “Estudiantes”.



Manual de Usuario Gráfico 23 Menú Usuario Fuente: Sebastián Troya

Profesores.

Este reporte nos permite observar en que cursos se encuentra un usuario con el rol de “Profesor”.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Profesor en Cursos

Profesor:

[Aceptar](#)

Cursos en los que se encuentra ADRIANA MONESTEROLO LENCIONI como Profesor

| Curso | Estado | Fecha de Creación | Última Fecha de Acceso al Curso |
|--|--------|-------------------|---------------------------------|
| GRUPOS SOCIALES CON DERECHOS ESPECÍFICOS (2011-01) | Activo | 2011-01-07 | 2011-09-02 |
| PROYECTO INTEGRADOR I (2011-01) | Activo | 2011-08-18 | 2011-09-02 |

Manual de Usuario Gráfico 24 Sub-Menú Profesores Fuente: Sebastián Troya

Estudiantes

Este reporte nos permite observar en que cursos se encuentra un usuario con el rol de “Estudiante”.

Estudiantes en Cursos

Estudiante:

[Aceptar](#)

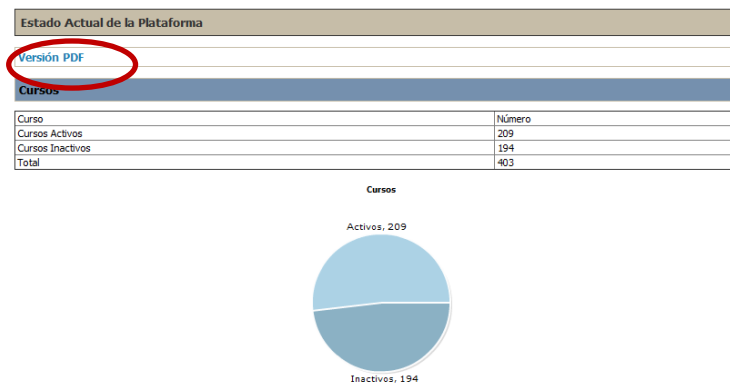
Cursos en los que se encuentra ABEL RODRIGO CABRERA MAXSI como Estudiante

| Curso | Estado | Fecha de Creación | Última Fecha de Acceso |
|--|----------|-------------------|------------------------|
| ELEC: EL SACERDOCIO EN EL CINE (2010-02) | Inactivo | 2011-02-14 | Nunca |

Manual de Usuario Gráfico 25 Sub-Menú Estudiantes Fuente: Sebastián Troya

Exportación a PDF.

Para visualizar los reportes en un documento “PDF” con el fin de poder imprimirlos se hace clic en el link “Versión PDF”.



Manual de Usuario Gráfico 26 Exportación a PDF Fuente: Sebastián Troya

Manual Técnico.

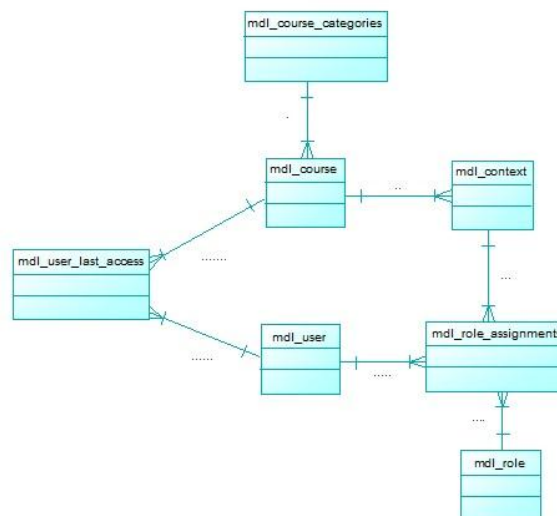
Introducción.

El presente manual presenta el diseño de base de datos con la cual se construyó el presente módulo.

Contiene un plan de contingencias en el caso de presentarse problemas de configuración luego de realizar cambios en la plataforma principal.

Modelo de Base de Datos

MOODLE posee más de 200 tablas en su base de datos, de las cuales se utilizaron Para la realización del módulo se requirió hacer consultas sobre tablas de la base de datos de MOODLE. En el presente gráfico se muestran solamente las tablas que nos interesan, ya que MOODLE posee más de 200 tablas. La dificultad se presenta en la no visualización de las relaciones entre las tablas. Pero después de hacer un estudio se encontró la relación entre las tablas que interesan para el módulo.

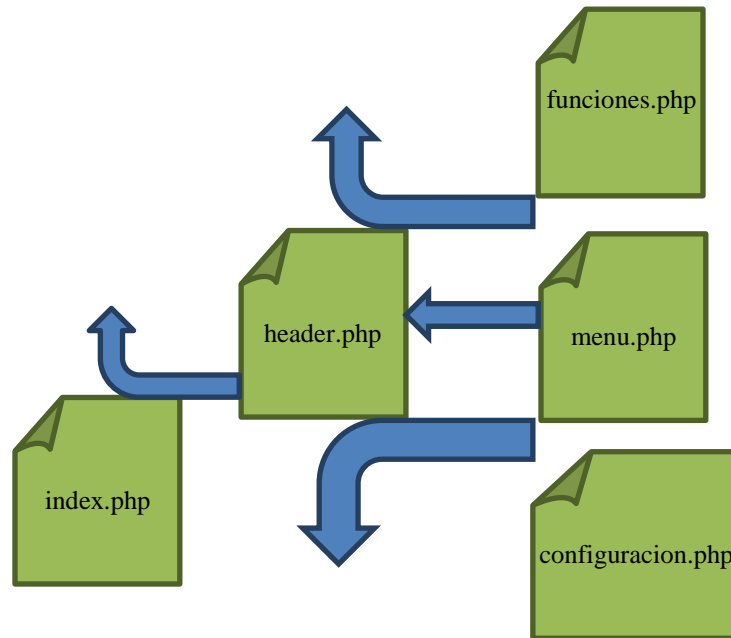


Diccionario de Datos.

- mdl_course_categories: tabla de categorías de la plataforma.
- mdl_course: tabla de cursos de la plataforma y a que categoría pertenece.
- mdl_user: tabla de usuarios de la plataforma.
- mdl_user_lastaccess: tabla donde se almacena el último acceso de un usuario a un curso.
- mdl_role: tabla de roles de la plataforma.
- mdl_context: tabla de los contextos de los curso de la plataforma
- mdl_role_assignments: tabla donde se almacena el usuario con el rol que tiene en un contexto.

Estructura del Módulo.

Para que un módulo funcione en MOODLE únicamente necesita dos archivos “index.php” y “settings.php”. El módulo para que tenga un mayor entendimiento se separó por partes los archivos. Es decir la parte gráfica se separó de los archivos de configuración etc.



Manual Técnico Gráfico 2 Estructura del Módulo Fuente: Sebastián Troya

Settings.php

El archivo “settings.php” define la configuración del módulo. En este archivo se define en que parte del árbol del menú se encuentra el módulo, el nombre del módulo, y la dirección física donde se encuentra el “index.php” del módulo.

```

<?php // $Id: settings.php,v 1.1.2.2 2008/11/26 20:58:04 skodak Exp $
$ADMIN->add("reports", new admin_externalpage('reportespuce','Reportes PUCE', "{$CFG->wwwroot}/{$CFG->admin/report/reportespuce/index.php",
'moodle/site:config'));
?>
  
```

Manual Técnico Gráfico 3 settings.php Fuente: Sebastián Troya

Errores.

El módulo no necesita ser configurado para que funcione, pero se pueden presentar errores. A continuación se presentan los errores más comunes que pueden suceder por hacer cambios en el módulo.

Objeto no encontrado al acceder al módulo por primera vez.



Manual Técnico Gráfico 4 Error 1 Fuente: Sebastián Troya

Este error se produce porque no existe el mismo nombre de la carpeta del módulo con la que se define en el archivo “settings.php”. Para corregir este error hay que tener una concordancia entre el nombre de la carpeta del módulo y las direcciones en el archivo “settings.php”.

```
<?php // $Id: settings.php,v 1.1.2.2 2008/11/26 20:55:44 skodak Exp $
$ADMIN->add("reports", new admin_externalpage('reportespuce','Reportes PUCE', "{$CFG->wwwroot}/{$CFG->admin}/report/reportespuce/index.php",
'moodle/site:config'));
?>
```

Manual Técnico Gráfico 5 Error 1-Solución Fuente: Sebastián Troya

Los parámetros que se encuentran en los círculos deben tener el mismo nombre de la dirección física en la que se encuentra el módulo.

Objeto no encontrado al entrar a un menú del módulo.



Manual Técnico Gráfico 6 Error 2 Fuente: Sebastián Troya

Este error se produce porque no existe el mismo nombre de la carpeta del módulo con la que se define en el archivo “menu.php”. Para corregir este error hay que tener una concordancia entre el nombre de la carpeta del módulo y las direcciones en el archivo “menu.php”.

```
<link href="css/style.css" type="text/css" rel="stylesheet" />
<link href="css/calendar.css" type="text/css" rel="stylesheet" />
<script src="js/jquery.js" type="text/javascript"></script>
<script src="js/funciones.js" type="text/javascript"></script>
<script src="js/calendar_us.js" type="text/javascript"></script>

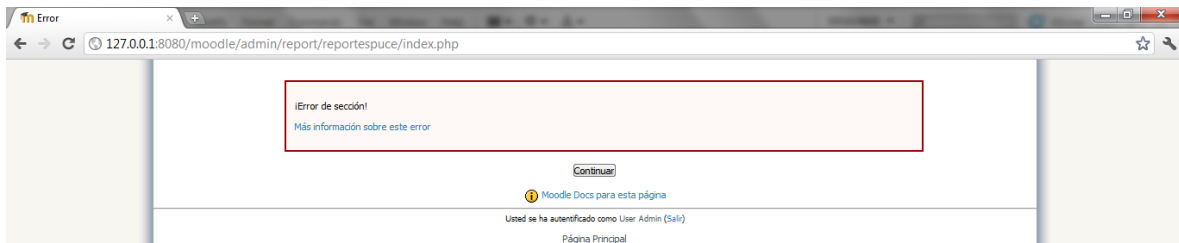
<?php $path=$CFG->wwwroot."/admin/report/reportespuce/">

<ul class="menu">
  <li><a href="#">General</a>
  <ul>
    <li><a href="#">Actual</a>
```

Manual Técnico Gráfico 7 Error2-Solución Fuente: Sebastián Troya

Los parámetros que se encuentran en los círculos deben tener el mismo nombre de la dirección física en la que se encuentra el módulo.

Error de Sección.



Manual Técnico Gráfico 8 Error 3 Fuente: Sebastián Troya

Este error se produce porque no existe el mismo nombre de la carpeta del módulo con la que se define en el archivo “header.php”. Para corregir este error hay que tener una concordancia entre el nombre de la carpeta del módulo y las direcciones en el archivo “header.php”.

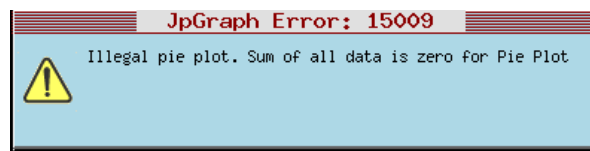
```
<?php
require_once('../.../config.php');
require_once($CFG->libdir.'/adminlib.php');
include("funciones.php");
admin_externalpage_setup('eportespuce');
admin_externalpage_print_header();
include('menu.php');
include("configuracion.php");

?>
```

Manual Técnico Gráfico 9 Error 3-Solución Fuente: Sebastián Troya

Los parámetros que se encuentran en los círculos deben tener el mismo nombre de la dirección física en la que se encuentra el módulo.

No se genera reporte PDF (“error 15009”).



Manual Técnico Gráfico 10 Error 4 Fuente: Sebastián Troya

Este error se produce porque no hay datos para graficar.

No se genera reporte PDF (“error 25111”).



Manual Técnico Gráfico 11 Error 5 Fuente: Sebastián Troya

Este error se produce porque se quedaron grabados en el servidor los gráficos del reporte. Para solucionar este problema vaya a la carpeta “reportes” que se encuentra dentro del módulo y borre todos los archivos con extensión “-png” que se encuentre allí.

Estándares de MOODLE

Antes de realizar algún código debemos tomar en cuenta las reglas que propone MOODLE para los desarrolladores. Estas reglas se tomaron de la página oficial de MOODLE. http://docs.moodle.org/19/es/Manual_de_Estilo_de_C%C3%B3digo

Reglas Generales.

1. Todos los ARCHIVOS DE CÓDIGO deberían utilizar la extensión `.php`.
2. Todas las PLANTILLAS deberían utilizar la extensión `.html`.
3. Todos los ARCHIVOS DE TEXTO deberían utilizar el formato de texto Unix (la mayoría de los editores de texto tienen esto como una opción).
4. Todas las ETIQUETAS php deben ser 'completas' como `<?php ?>` ... no 'reducidas' como `<? ?>`.
5. Todos los AVISOS DE COPYRIGHT deben ser mantenidos. Puede incluir los suyos propios si resulta necesario.
6. Todos los ARCHIVOS deben incluir el archivo principal `config.php`.
7. Cualquier otro `include/require` debería utilizar una RUTA absoluta que comience por `$CFG->dirroot` o `$CFG->libdir`, nunca relativos, ya que estos en algunas ocasiones funcionan de forma extraña en PHP.
8. Cada archivo debería comprobar que el usuario está AUTENTICADO correctamente, utilizando las funciones `require_login()` y `isadmin()`, `isteacher()`, `iscreator()` o `isstudent()`.
9. Todos los **ACCESOS A LA BASE DE DATOS** deberían utilizar las funciones definidas en `lib/datalib.php` cuando sea posible - esto permite la compatibilidad con un gran número de bases de datos. Debería encontrar que prácticamente todo es posible

utilizando estas funciones. Si quiere escribir código SQL entonces deberá comprobar que: funciona en cualquier plataforma; restringido a funciones específicas de su código (normalmente un archivo lib.php); y claramente comentado.

10. No cree o utilice **VARIABLES** globales distintas de las estándar \$CFG, \$SESSION, \$THEME, \$SITE, \$COURSE y \$USER.

11. Todas las **VARIABLES** deberían ser inicializadas o, al menos, comprobada su existencia utilizando isset() o empty() antes de ser utilizadas.

12. Todas las **CADENAS** deberían ser traducibles - cree nuevos textos en los archivos "lang/es_utf8" con palabras reducidas en inglés y su traducción completa al Español y recupérelas en su código utilizando las funciones get_string() or print_string().

13. Todos los **ERRORES** deberían ser visualizados utilizando la función print_error() para maximizar la traducción y ayudar a los usuarios (Automáticamente se enlaza con Moodle Docs).

14. Todos los **FICHEROS DE AYUDA** deben ser traducibles - cree nuevos textos en el directorio "lang/es_utf8/help" y llámelos utilizando la función helpbutton().

15. 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 puedes 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.

16. Todos los textos dentro de Moodle, especialmente aquellos que han sido introducidos por los usuarios, deben ser mostrados utilizando la **FUNCIÓN** format_text(). Esto asegura que el texto es filtrado y limpiado correctamente.

17. Las **ACCIONES DE LOS USUARIOS** deberían ser grabadas utilizando la función `add_to_log()`. Estos registros son utilizados para la generación de los "Informes de Actividad" y los Registros.

18. Al generar **ENLACES HTML**, hágalos siempre relativos a la raíz del sitio Moodle, por ejemplo, enlace a `$CFG->wwwroot/mod/blonk/view.php?id=99` en lugar de únicamente `view.php?id=99`. Esto causa que su código funcionará aunque sea llamado por un script que se encuentre en otra carpeta diferente.

Estilo del Código.

El sangrado del texto debe ser siempre de 4 espacios. No utilices los tabuladores NUNCA.

```
1 <?php
2 require('config.php');
3 $a = required_param('a', PARAM_INT);
4 if ($a > 10) {
5     call_some_error($a);
6 } else {
7     do_something_with($a);
8 }
9 ?>
```

Anexos Estándares Moodle 1 Código bien escrito Fuente: Sebastián Troya

```
1 <?php
2     require('config.php');
3     $a = required_param('a', PARAM_INT);
4     if ($a > 10) {
5         call_some_error($a);
6     } else {
7         do_something_with($a);
8     }
9     ?>
```

Anexos Estándares Moodle 2 Código mal escrito Fuente: Sebastián Troya

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

```
<?php
BIEN: $quiz
BIEN: $errorstring
BIEN: $assignments (for an array of objects)
BIEN: $i (but only in little loops)
?>
```

```
<?php
MAL: $Quiz
MAL: $aReallyLongVariableNameWithoutAGoodReason
MAL: $error_string
?>
```

Anexos Estándares Moodle 3 Declaración de Variables Fuente: Sebastián Troya

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.

```
<?php
define("FORUM_MODE_FLATOLDEST", 1);
?>
```

Anexos Estándares Moodle 4 declaración de Constantes Fuente: Sebastián Troya

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

```
<?php
function forum_set_display_mode($mode=0) {
    global $USER, $CFG;
    if ($mode) {
        $USER->mode = $mode;
    } else if (empty($USER->mode)) {
        $USER->mode = $CFG->forum_displaymode;
    }
}
?>
```

Anexos Estándares Moodle 5 Declaración de Funciones Fuente: Sebastián Troya

Los bloques de código siempre deben estar encerrados por llaves (incluso si solo constan de una línea).

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

Anexos Estándares Moodle 6 Bloques de Código Fuente: Sebastián Troya

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

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

Anexos Estándares Moodle 7 Cadenas de texto Fuente: Sebastián Troya

Los comentarios deben ser añadidos de forma que resulten prácticos, para explicar el flujo del código y el propósito de las funciones y variables.

- Cada función (y cada clase) debería utilizar el popular formato phpDoc. Esto permite que la documentación sea generada automáticamente.

- Los comentarios en línea deberían utilizar los caracteres //, alineados con cuidado por encima de las líneas de código que comenta.

```
<?php
/**
 * 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
            }
        }
    }
}
?>
```

Anexos Estándares Moodle 8 Comentarios Fuente: Sebastián Troya

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

```
<?php
MAL:  $b = $a;
BIEN: $b = clone($a);
?>
```

Anexos Estándares Moodle 9 Copia de Variables Fuente: Sebastián Troya

Estructura de la base de Datos.

- Cada tabla debe tener un campo auto numérico id (INT10) como clave primaria.
- La tabla principal que contiene instancias de cada módulo debe tener el mismo nombre que el módulo y contener, por lo menos, los siguientes campos:
 - Id.
 - Course.
 - Name.
- El resto de las tablas asociadas con un módulo que contiene información sobre 'cosas', deberían ser llamadas modulo_cosas (fíjese en el plural!).
- Los nombres de las tablas y de los campos tienen que evitar el uso de palabras reservadas por las Bases de Datos. Por favor, compruébelo antes de crearlas.
- Los nombres de los campos (columnas) deberían ser sencillos y cortos, siguiendo las mismas reglas que los nombres de las variables.
- Cuando sea posible, las columnas que contengan una referencia al campo id de otra tabla (por ejemplo, modulo) debería ser llamado moduloid. (fíjate que esta norma es nueva y no es seguida por algunas tablas antiguas).
- Los campos booleanos serán implementados como enteros cortos (por ejemplo, INT4) con los valores 0 o 1, para permitir la futura expansión de los valores si fuera necesario.
- La mayoría de las tablas tienen que tener un campo timemodified (INT10) que será actualizado con la fecha actual (timestamp de UNIX) obtenida con la función time() de PHP.
- Defina siempre un valor por defecto para cada campo (y haga que tenga sentido).

- Cada tabla debe comenzar con el prefijo de la base de datos (\$CFG->prefix). En muchos casos esto es gestionado automáticamente. Además, bajo PostgreSQL, el nombre de cada índice debe empezar también con el prefijo.
- Para garantizar la compatibilidad entre bases de datos, por favor, siga las reglas siguientes sobre el uso del comando **AS** (solo si necesita alias en tablas/campos, por supuesto):
 - **No utilice** el comando **AS** para **alias de tablas**.
 - **Utilice** el comando **AS** para **alias de campos (columnas)**.
- **Nunca** cree UNIQUE KEY (restricciones) para nada. En su lugar utilice UNIQUE INDEX. En el futuro, si se decide añadir integridad referencial a Moodle y si se necesitan UNIQUE KEY, serán utilizadas, pero no por ahora. Por favor, fíjese que el Editor XMLDB permite especificar tanto restricciones UNIQUE y FOREIGN (y eso es bueno, teniendo el XML bien definido), pero solo los índices subyacentes serán realmente generados en la DB.
- Esas UNIQUE KEYs creadas en el Editor XMLDB (lea el punto anterior) solo debe ser definida si el campo/campos **van a ser el objetivo** para alguna FOREIGN KEY (a nivel de Editor). En caso contrario, creelas como UNIQUE INDEXes.
- Las tablas asociadas **con un bloque** deben seguir las siguientes convenciones en sus nombres: **\$CFG->prefix + "block_" + nombre del bloque + añadidos**. Por ejemplo, asumiendo que \$CFG->prefix es 'mdl_', todas las tablas para el bloque "rss_client" deberán empezar por 'mdl_block_rss_client' (siendo posible añadir más palabras al final, p.ej. 'mdl_block_rss_client_anothertable!...'). Esta regla será

completamente forzada con Moodle 2.0, dando algo de tiempo a los desarrolladores hasta entonces.

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

Normas de Seguridad (y control de la información de formularios y URLs)

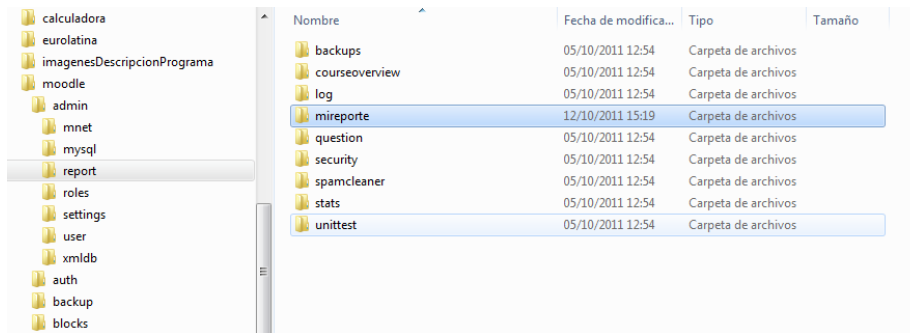
1. No se base en `'register_globals'`. Cada variable debe ser correctamente inicializada en cada fichero de código. Debe ser obvia la procedencia de cada variable.
2. Inicialice todos los arrays y objetos aunque estén vacíos. `$a = array()` o `$obj = new stdClass();`.
3. No utilice la función `optional_variable()`. En su lugar, utilice la función `optional_param()`. Seleccione la opción `PARAM_XXXX` apropiada al tipo de parámetro que espera. Para comprobar y definir un valor opcional para una variable, utilice la función `set_default()`.
4. No utilice la función `require_variable()`. En su lugar, utilice la función `required_param()`. Seleccione la opción `PARAM_XXXX` apropiada al tipo de parámetro que espera.
5. Utilice `data_submitted()`, con cuidado. La información todavía debe ser limpiada antes de utilizarla.

6. No utilice `$_GET`, `$_POST` o `$_REQUEST`. En su lugar, utilice las funciones `required_param()` o `optional_param()` apropiadas.
7. No compruebe las acciones con código como: `if (isset($_GET['algo']))`. Utilice, por ejemplo, `$algo = optional_param(algo, -1, PARAM_INT)` y entonces compruebe que está dentro de los valores esperados, por ejemplo, `if ($something>=0) {....`
8. Cuando sea posible agrupe todas sus llamadas a `required_param()`, `optional_param()` y el resto de inicialización de variables en el principio de cada fichero (o función) para que sea fácilmente localizable.
9. Utilice el mecanismo 'sesskey' para proteger el envío de formularios de ataques. Un ejemplo de uso: cuando el formulario es generado, incluya `<input type="hidden" name="sesskey" value="<?php echo sesskey(); ?>" />`. Cuando el formulario es procesado, compruebe `if (!confirm_sesskey()) {error('Bad Session Key');}`.
10. Todos los nombres de ficheros deben ser 'limpiados' utilizando la función `clean_filename()`, si esto no ha sido realizado con el uso de las funciones `required_param()` o `optional_param()` con anterioridad.
11. Cualquier información leída desde la base de datos debe tener la función `addslashes()` aplicada antes de volver a enviar la información a la base de datos. Un objeto completo puede ser procesado con la función `addslashes_object()`.
12. Cuando sea posible, la información que se almacenará en la base de datos debe venir de peticiones POST (por ejemplo, información de un formulario) en lugar de utilizar peticiones GET (por ejemplo, información de la URL).
13. No utilice información obtenida de `$_SERVER` si puede evitarlo. Presenta algunos problemas de portabilidad.

14. Si no ha sido realizado en ningún otro lugar, asegúrese de que la información enviada a la base de datos ha sido filtrada mediante la función `clean_param()` utilizando la opción `PARAM_XXXX` apropiada.
15. Si escribe código SQL, asegúrese completamente de que es correcto. En particular, compruebe la falta de comillas en las variables utilizadas. Es un punto de entrada para ataques de tipo 'SQL injection'.
16. Compruebe toda la información (especialmente la que es enviada a la base de datos) en cada archivo que es utilizada. Nunca confíe en que otro código estará haciendo ese trabajo.
17. Los bloques de código que se incluyan deben presentar una estructura PHP correcta (por ejemplo, con una declaración de una clase, de funciones, etc.) - los bloques de código lineales ("espagueti") suelen tender a utilizar variables sin inicializar (y son menos legibles).
18. Si necesita usar `shell_exec()` (o cualquier otra función que invoque un *shell*), asegúrese de que ha limpiado los parámetros anteriormente con `escapeshellcmd()/escapeshellarg()` (de lo contrario abrimos la puerta a ataques de inyección de *shell*).

Como hacer un módulo en MOODLE.

Lo primero que hay que tomar en cuenta es el contexto que se desea que el módulo se ejecute. Como este módulo será solamente vista por el administrador y la función principal será la de hacer reportes, la ubicación de los archivos se los debe hacer en la carpeta “admin/report/”.



Anexos Como hacer un módulo 1 Ubicación del Módulo Fuente: Sebastián Troya

Para que nuestro módulo funciones necesita sólo y obligatoriamente de dos archivos: index.php y settings.php. Con estos dos archivos se le dice al core de Moodle que nuestro módulo está listo para mostrarse.

Settings.php

Este es el archivo más importante ya que en este archivo se hace la configuración del módulo. El archivo sólo tiene una línea de código.

```
<?php // $Id: settings.php,v 1.1.2.2 2008/11/26 20:58:04 skodak Exp $
$ADMIN->add("reports", new admin_externalpage('mireporte', 'Mis Reportes', "$CFG->wwwroot/$CFG->admin/report/mireporte/index.php",
"moodle/site:viewreports", 0));
?>
```

Anexos Como hacer un módulo 2 Settings.php Fuente: Sebastián Troya

Como este módulo será desplegado en el bloque de administración del Administrador, lo primero es llamar al objeto \$ADMIN que es una instancia de “admin_root” que es la clase que maneja al administrador del sitio.

Se llama al método “add” de la clase “admin_root” que lo que hace es añadir un ítem en el árbol del bloque de administración. Este método recibe 2 parámetros.

- Padre.- Es bajo que categoría del bloque se muestra.
- Algo.- Es una instancia de un objeto o una parte del árbol de administración.

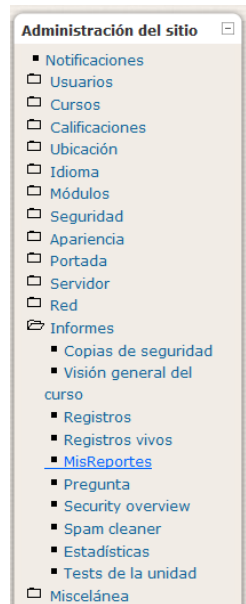
En el código anterior vemos que el padre es “reports” es decir se va a mostrar en el menú “reportes” una página externa “admin_externalpage”.

admin_externalpage

Este objeto permite insertar una página externa en el árbol de administración. El constructor recibe 6 variables: (lib/adminlib.php. línea 1287)

- Name: Es el nombre interno de la página externa. Debe ser única.
- Visiblename: Muestra el nombre de la página externa. Usualmente se la obtiene a través de la función get_string().
- url: La dirección externa de la página externa. Debemos recordar las reglas para generar el path.
- req_capability: Son los permisos que se les da al rol del usuario para tener acceso del usuario a la página externa. Por defecto se define ‘moodle/site:config’ (<http://docs.moodle.org/20/en/Category:Capabilities>)
- hidden: Es la visibilidad de la página externa en el árbol del menú.

- Context: Es el contexto de la página externa.



Anexos Como hacer un módulo 3 Árbol de Administración Fuente: Sebastián Troya

Index.php

El archivo index.php debe tener líneas de código obligatorias que las veremos a continuación.

```
1 <?php // $Id: index.php,v 1.2.2.5 2010/09/08 16:51:29 stronk7 Exp $
2 // index.php - scheduled backup logs
3
4 require_once('../../../config.php');
5 require_once($CFG->libdir.'/adminlib.php');
6 admin_externalpage_setup('mireporte');
7 admin_externalpage_print_header();
8 print("esta es una prueba");
9 print_footer();
10 ?>
```

Anexos Como hacer un módulo 4 index.php Fuente: Sebastián Troya

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

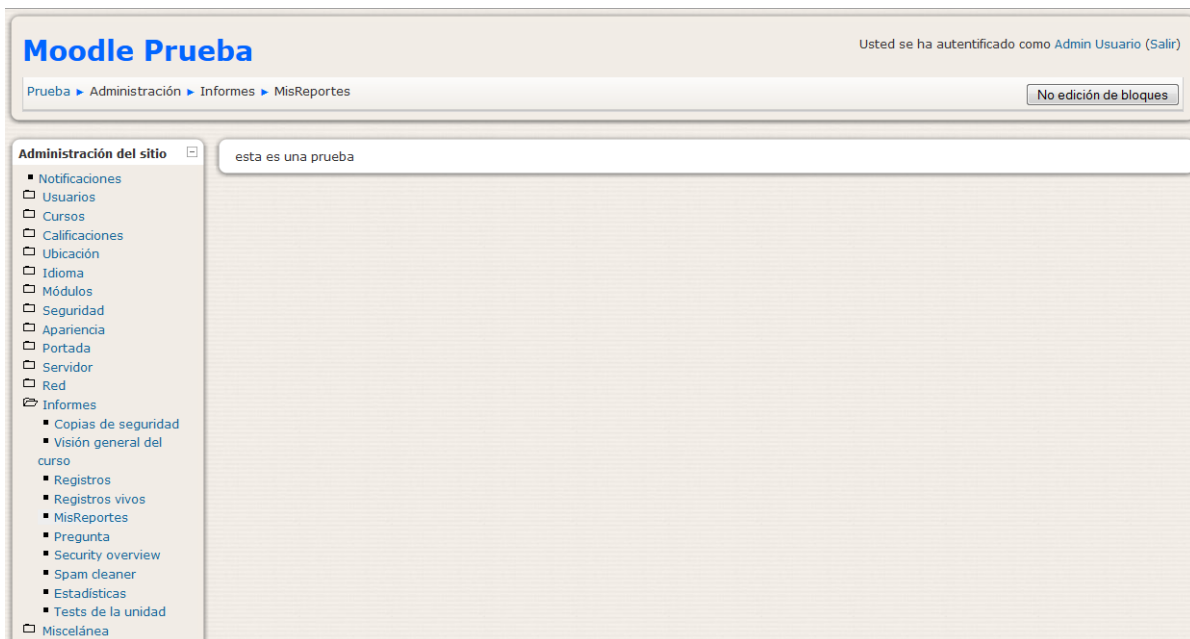
Debe incluir dos archivos: el archivo 'config.php' y 'adminlib.php' (líneas 4 y 5).

La función 'admin_externalpage_setup' recibe como parámetro el 'name' que definimos en el archivo 'settings.php' y lo que hace es 'setear' a la página externa. (línea 6).

Se llama a la función 'admin_externalpage_print_header' para mostrar la cabecera del tema. (línea 7).

Con la función 'print' se muestra en la pantalla lo que deseamos (Línea 8).

Llamamos a la función 'print_footer' para imprimir el pie del tema. (Línea 9).



Anexos Como hacer un módulo 5 Pantalla Módulo.php Fuente: Sebastián Troya

Bibliografía

Ingeniería de Requisitos en Aplicaciones para la web -Estudio comparativo . (Diciembre de 2002). *Ingeniería de Requisitos en Aplicaciones para la web -Estudio comparativo*. Sevilla, España: Universidad de Sevilla.

Introducción Ingeniería de Software. (31 de Octubre de 2007). Recuperado el 11 de Octubre de 2011, de Introducción Ingeniería de Software:
<http://xherrera334.blogspot.es/1193789220/>

Wikipedia. (30 de Diciembre de 2011). Recuperado el 3 de Enero de 2012, de Wikipedia:
<http://es.wikipedia.org/wiki/Modelo>

Alba, M. J. (Diciembre de 2010). Análisis del desarrollo de extensiones para Moodle: Desarrollo de un módulo para la gestión de laboratorios docentes. *Análisis del desarrollo de extensiones para Moodle: Desarrollo de un módulo para la gestión de laboratorios docentes*. Alcalá, España.

Campos, O. (29 de Septiembre de 2011). *GENBETA:dev*. Recuperado el 13 de Octubre de 2011, de GENBETA:dev: <http://www.genbetadev.com/desarrollo-web/escribiendo-modulos-para-moodle-introduccion>

Departamento de Nuevas Tecnologías. (2010). *Estudio del Uso de las TICs en la PUCE*. Quito.

Dirección de Informática PUCE. (s.f.). *PuceVirtual*. Recuperado el 2011 de Mayo de 2011, de <http://www.puce.edu.ec/pucevirtual/index.php>

epestudio. (16 de Agosto de 2005). *Especialistas Web*. Recuperado el 11 de Octubre de 2011, de Especialistas Web: <http://www.epestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm>

Maestros del web. (s.f.). Recuperado el 6 de Octubre de 2011, de <http://www.maestrosdelweb.com/editorial/introcss/>

Moodle. (17 de Julio de 2008). *Moodle*. Recuperado el 13 de Octubre de 2011, de Moodle: http://docs.moodle.org/19/es/Manual_de_Estilo_de_C%C3%B3digo

Moodle. (2011 de Octubre de 2011). *Moodle*. Recuperado el 13 de Octubre de 2011, de Moodle: http://docs.moodle.org/20/en/Roles_and_capabilities

PHP. (7 de Octubre de 2011). *PHP*. Recuperado el 7 de Octubre de 2011, de PHP: <http://www.php.net/manual/es/preface.php>

Recursos Gráficos. (29 de Diciembre de 2007). *Recursos Gráficos*. Recuperado el 6 de Octubre de 2011, de Recursos Gráficos: <http://recursosgraficos.espinlog.com/?p=47>

Slideshare. (s.f.). *SlideShare*. Recuperado el 11 de Octubre de 2011, de SlideShare: <http://www.slideshare.net/inventa2/modelos-de-desarrollo>

Terminología de E-learning. (s.f.). *PuceVirtual*. Recuperado el 17 de Mayo de 2011, de <http://www.puce.edu.ec/pucevirtual/elearning.htm>

Troya, P. S.-S. (2008). SOFTWARE PARA AUTOMATIZACIÓN DEL PROCESO DE REDUCCIÓN DE PUNTOS AL CONDUCTOR. *SOFTWARE PARA AUTOMATIZACIÓN DEL PROCESO DE REDUCCIÓN DE PUNTOS AL CONDUCTOR*. Quito, Pichincha, Ecuador.

Wikipedia. (28 de Diciembre de 2011). *Wikipedia*. Recuperado el 3 de Enero de 2012, de <http://es.wikipedia.org/wiki/PHP>

Wikipedia. (18 de Septiembre de 2011). *Wikipedia*. Recuperado el 2011 de Octubre de 6, de Wipkipedia: http://es.wikipedia.org/wiki/Adobe_Dreamweaver

Wikipedia. (16 de Septiembre de 2011). *Wikipedia*. Recuperado el 7 de Octubre de 2011, de Wikipedia: <http://es.wikipedia.org/wiki/PHP>

Índice de Ilustraciones.

Grafico1 1 Uso de Plataformas por parte de los docentes de la PUCE Fuente: Departamento de Nuevas Tecnologías 3

Gráfico 2 1 Ciclo de Vida Modelo Iterativo Incremental Fuente: <http://xherrera334.blogspot.es> 14

Gráfico 2 2 Estructura Básica de MOODLE Fuente: Sebastián Troya 20

Gráfico 2 3 Modelo de base de datos Moodle 1.9 Fuente: Sebastián Troya 24

Gráfico 3 1 Prototipo Menú Principal Fuente: Sebastián Troya 34

Gráfico 3 2 Prototipo General Fuente: Sebastián Troya..... 35

Gráfico 3 3 Prototipo Sitio Fuente: Sebastián Troya..... 35

Gráfico 3 4 Prototipo Curso Fuente: Sebastián Troya..... 35

Gráfico 3 5 Prototipo Usuarios Fuente: Sebastián Troya 36

Gráfico 3 6 Diseño de Alto Nivel Fuente: Sebastián Troya 36

Gráfico 4. 1 Diagrama de Casos de Uso. Fuente: Sebastián Troya..... 37

Gráfico 4. 2 Escenario F1. Fuente: Sebastián Troya 39

Gráfico 4. 3 Escenario F1.1.1. Fuente: Sebastián Troya 40

Gráfico 4. 4. Escenario F1.1.2. Fuente: Sebastián Troya 41

Gráfico 4. 5. Escenario F1.2.1. Fuente: Sebastián Troya 42

Gráfico 4. 6. Escenario F1.2.2. Fuente: Sebastián Troya 43

| | |
|---|----|
| Gráfico 4. 7. Escenario F1.2.3. Fuente: Sebastián Troya | 44 |
| Gráfico 4. 8. Escenario F1.3.1. Fuente: Sebastián Troya | 45 |
| Gráfico 4. 9. Escenario F1.3.2. Fuente: Sebastián Troya | 46 |
| Gráfico 4. 10. Escenario F2. Fuente: Sebastián Troya | 49 |
| Gráfico 4. 11. Escenario F2.1. Fuente: Sebastián Troya | 50 |
| Gráfico 4. 12. Escenario F2.2. Fuente: Sebastián Troya | 51 |
| Gráfico 4. 13. Escenario F3. Fuente: Sebastián Troya | 54 |
| Gráfico 4. 14. Escenario F3.1. Fuente: Sebastián Troya | 55 |
| Gráfico 4. 15. Escenario F3.2. Fuente: Sebastián Troya | 56 |
| Gráfico 4. 16. Escenario F3.3.1. Fuente: Sebastián Troya | 57 |
| Gráfico 4. 17. Escenario F3.3.2. Fuente: Sebastián Troya | 58 |
| Gráfico 4. 18. Escenario F3.4.1. Fuente: Sebastián Troya | 59 |
| Gráfico 4. 19. Diagrama Entidad-Relación. Fuente: Sebastián Troya | 61 |
| | |
| Manual de Usuario Gráfico 1 Donde Instalar el Módulo Fuente: Sebastián Troya | 65 |
| Manual de Usuario Gráfico 2 Menú Fuente: Sebastián Troya | 66 |
| Manual de Usuario Gráfico 3 Menú General Fuente: Sebastián Troya | 66 |
| Manual de Usuario Gráfico 4 Submenu Actual Fuente: Sebastián Troya..... | 67 |
| Manual de Usuario Gráfico 5 General Actual Fuente: Sebastián Troya | 67 |
| Manual de Usuario Gráfico 6 Usuarios Sin Curso Fuente: Sebastián Troya | 68 |
| Manual de Usuario Gráfico 7 Cursos Más Visitados Fuente: Sebastián Troya | 68 |
| Manual de Usuario Gráfico 8 Sub-Menú Por Periodos Fuente: Sebastián Troya | 69 |
| Manual de Usuario Gráfico 9 Un solo Periodo Ingreso Datos Fuente: Sebastián Troya | 69 |

| | |
|--|----|
| Manual de Usuario Gráfico 10 Comparación Entre Periodos General Fuente: Sebastián Troya..... | 70 |
| Manual de Usuario Gráfico 11 Cursos más Visitados en un Periodo Fuente: Sebastián Troya..... | 71 |
| Manual de Usuario Gráfico 12 Menú Sitio Fuente: Sebastián Troya | 72 |
| Manual de Usuario Gráfico 13 Categorías Fuente: Sebastián Troya | 72 |
| Manual de Usuario Gráfico 14 Cursos en Categorías Fuente: Sebastián Troya | 73 |
| Manual de Usuario Gráfico 15 Recursos por Curso Fuente: Sebastián Troya..... | 74 |
| Manual de Usuario Gráfico 16 Estudiantes por Curso Fuente: Sebastián Troya | 74 |
| Manual de Usuario Gráfico 17 Sub-Menú Un solo Periodo Fuente: Sebastián Troya | 75 |
| Manual de Usuario Gráfico 18 Cursos Creados Un solo Periodo Parámetros Fuente: Sebastián Troya | 75 |
| Manual de Usuario Gráfico 19 Cursos Creados en un solo Periodo Fuente: Sebastián Troya | 76 |
| Manual de Usuario Gráfico 20 Tráfico del Curso en Un solo Periodo Fuente: Sebastián Troya..... | 76 |
| Manual de Usuario Gráfico 21 Comparación entre Periodos Fuente: Sebastián Troya..... | 77 |
| Manual de Usuario Gráfico 22 Creación de Cursos Comparación entre Periodos Fuente: Sebastián Troya | 77 |
| Manual de Usuario Gráfico 23 Menú Usuario Fuente: Sebastián Troya | 78 |
| Manual de Usuario Gráfico 24 Sub-Menú Profesores Fuente: Sebastián Troya..... | 79 |
| Manual de Usuario Gráfico 25 Sub-Menú Estudiantes Fuente: Sebastián Troya | 79 |
| Manual de Usuario Gráfico 26 Exportación a PDF Fuente: Sebastián Troya..... | 79 |

| | |
|---|----|
| Manual Técnico Gráfico 1 Modelo de Base de Datos Fuente: Sebastián Troya..... | 80 |
| Manual Técnico Gráfico 2 Estructura del Módulo Fuente: Sebastián Troya..... | 82 |
| Manual Técnico Gráfico 3 settings.php Fuente: Sebastián Troya..... | 82 |
| Manual Técnico Gráfico 4 Error 1 Fuente: Sebastián Troya..... | 83 |
| Manual Técnico Gráfico 5 Error 1-Solución Fuente: Sebastián Troya..... | 83 |
| Manual Técnico Gráfico 6 Error 2 Fuente: Sebastián Troya..... | 83 |
| Manual Técnico Gráfico 7 Error2-Solución Fuente: Sebastián Troya..... | 84 |
| Manual Técnico Gráfico 8 Error 3 Fuente: Sebastián Troya..... | 84 |
| Manual Técnico Gráfico 9 Error 3-Solución Fuente: Sebastián Troya..... | 85 |
| Manual Técnico Gráfico 10 Error 4 Fuente: Sebastián Troya..... | 85 |
| Manual Técnico Gráfico 11 Error 5 Fuente: Sebastián Troya..... | 85 |
| | |
| Anexos Estándares Moodle 1 Código bien escrito Fuente: Sebastián Troya..... | 88 |
| Anexos Estándares Moodle 2 Código mal escrito Fuente: Sebastián Troya..... | 88 |
| Anexos Estándares Moodle 3 Declaración de Variables Fuente: Sebastián Troya..... | 89 |
| Anexos Estándares Moodle 4 declaración de Constantes Fuente: Sebastián Troya..... | 89 |
| Anexos Estándares Moodle 5 Declaración de Funciones Fuente: Sebastián Troya..... | 90 |
| Anexos Estándares Moodle 6 Bloques de Código Fuente: Sebastián Troya..... | 90 |
| Anexos Estándares Moodle 7 Cadenas de texto Fuente: Sebastián Troya..... | 90 |
| Anexos Estándares Moodle 8 Comentarios Fuente: Sebastián Troya..... | 91 |
| Anexos Estándares Moodle 9 Copia de Variables Fuente: Sebastián Troya..... | 91 |
| | |
| Anexos Como hacer un módulo 1 Ubicación del Módulo Fuente: Sebastián Troya | 97 |

Anexos Como hacer un módulo 2 Settings.php Fuente: Sebastián Troya 97

Anexos Como hacer un módulo 3 Árbol de Administración Fuente: Sebastián Troya 99

Anexos Como hacer un módulo 4 index.php Fuente: Sebastián Troya 99

Anexos Como hacer un módulo 5 Pantalla Módulo.php Fuente: Sebastián Troya 100

Contenido

| | |
|--|----------|
| Capítulo I: Análisis del Módulo de seguimiento del uso de la Plataforma MOODLE. | 1 |
| 1.1 Introducción. | 1 |
| 1.2 Antecedentes | 2 |
| 1.2.1 ¿Cómo está actualmente el proyecto PuceMoodle? | 4 |
| 1.2.2 ¿Cómo se opera actualmente? | 5 |
| 1.2.3 Fuente del Problema. | 5 |
| Capítulo II: Herramientas para el desarrollo. | 6 |
| 2.1 Herramientas..... | 6 |
| 2.1.2 PHP | 8 |
| 2.1.4 MySql. | 10 |
| 2.2 Modelo de Desarrollo. | 11 |
| 2.2.1 Introducción..... | 11 |
| 2.2.2 Desarrollo Iterativo Incremental..... | 12 |
| 2.3 ¿Qué es MOODLE?..... | 17 |
| 2.3.1 Introducción..... | 17 |
| 2.3.2 Estructura de MOODLE..... | 20 |
| Capítulo III: Levantamiento de Requerimientos. | 25 |
| 3.1 ¿Qué son los Requerimientos? | 25 |
| 3.2 Métodos para levantar Requerimientos. | 26 |

| | |
|---|----|
| 3.3 Variables a dar seguimiento. | 30 |
| 3.4 Catálogo de Requerimientos..... | 31 |
| 3.4.1 Antecedentes..... | 31 |
| 3.4.2 Estrategia. | 32 |
| 3.4.3 Requerimientos Funcionales | 32 |
| 3.4.4 Requisitos No Funcionales..... | 33 |
| 3.4.5 Requerimientos de Funcionamiento..... | 34 |
| 3.4.6 Prototipo. | 34 |
| 3.4.7 Diseño de Alto Nivel. | 36 |
| Capítulo IV: Desarrollo del Módulo..... | 37 |
| 4.1 Diagramas de Caso de Uso..... | 37 |
| 4.2 Actor. | 38 |
| 4.3 Desarrollo del Módulo..... | 38 |
| 4.3.1 Primer Ciclo..... | 38 |
| 4.3.2 Segundo Ciclo. | 48 |
| 4.3.3 Tercer Ciclo. | 53 |
| 4.4 Diagrama Entidad Relación..... | 61 |
| Capítulo V: Conclusiones y Recomendaciones..... | 62 |
| 5.1 Conclusiones..... | 62 |
| 5.2 Recomendaciones. | 63 |

| | |
|--|-----|
| Anexos..... | 65 |
| Manual de Usuario. | 65 |
| Introducción..... | 65 |
| Instalación..... | 65 |
| Requerimientos de Funcionamiento. | 66 |
| Funcionamiento. | 66 |
| Exportación a PDF. | 79 |
| Manual Técnico..... | 80 |
| Introducción..... | 80 |
| Modelo de Base de Datos | 80 |
| Diccionario de Datos. | 81 |
| Estructura del Módulo. | 81 |
| Settings.php | 82 |
| Errores. | 82 |
| Error de Sección. | 84 |
| Estándares de MOODLE..... | 86 |
| Estilo del Código. | 88 |
| Estructura de la base de Datos..... | 92 |
| Como hacer un módulo en MOODLE..... | 97 |
| Bibliografía..... | 101 |

Índice de Ilustraciones..... 104