

**Sistema de administración de contenidos**  
*El cuarto de lavado*

**Proyecto de grado**

**Hernando Andrés Silva Vega**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS**  
**FACULTAD DE INGENIERÍA**  
**PROGRAMA TECNOLOGÍA EN INFORMÁTICA Y**  
**TELECOMUNICACIONES**  
**Bogota D.C.**  
**Enero de 2008**

**Sistema de administración de contenidos**  
*El cuarto de lavado*

**Proyecto de grado**

**Hernando Andrés Silva Vega**

**Asesor:**  
**Manuel Moreno**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS**  
**FACULTAD DE INGENIERÍA**  
**PROGRAMA TECNOLOGÍA EN INFORMÁTICA Y**  
**TELECOMUNICACIONES**  
**Enero de 2008**

PAGINA DE ACEPTACION

NOTA DE ACEPTACION

---

---

---

---

---

---

PRESIDENTE DEL JURADO

---

JURADO

---

JURADO

## Dedicatoria

Quiero dedicar este trabajo a las personas que invirtieron en mí y hace mucho esperan. A todos aquellos que por la razón que fuere se lanzan solitarios donde nadie se atreve. A quienes una vez solos creen y hacen. A quienes a si mismos se proveen el significado para no ahogarse.

## Agradecimientos

Todo proyecto de grado, por ser un trabajo de consumación de varios años, es extenso y dispendioso. Además es necesario acudir de nuevo a todo el conocimiento por el que pasamos durante el recorrido y esforzarnos por hacer que todo ello que aprendimos tan aparte y con sus respectivos ejemplos propios, funcione junto como un relojito con la promesa de que todo ello no tiene pierda. Por eso agradezco a todos ellos que me han dado la mano en el proceso de construcción de este trabajo y más aun a todos ellos que han apretado mi corazón para que camine con mis propias piernas y vivir con el orgullo de ver materializada la voluntad.

## Contenido

Contenido .....	6
Índice de tablas.....	8
Índice de gráficos.....	8
INTRODUCCION .....	9
2 TEMA.....	10
2.1 TÍTULO DEL PROYECTO.....	10
2.2 DESCRIPCIÓN DEL PROBLEMA.....	11
2.3 JUSTIFICACIÓN .....	12
2.4 ANTECEDENTES.....	12
2.5 DESCRIPCIÓN DEL SISTEMA ACTUAL .....	12
2.6 OBJETIVOS .....	14
2.6.1 Objetivo general. ....	14
2.6.2 Objetivos específicos.....	14
2.7 METODOLOGÍA DE INVESTIGACIÓN.....	14
2.7.1 Línea de investigación.....	15
2.8 ESTADO DEL ARTE .....	16
2.9 MARCO TEÓRICO .....	18
2.9.1 PHP .....	19
2.9.2 HTML.....	20
2.9.3 SQL.....	20
2.9.4 JavaScript.....	21
2.9.5 Software libre.....	21
2.9.6 Formatos de archivo a utilizar .....	22
3 INGENIERÍA DEL PROYECTO .....	26
3.1 METODOLOGÍA Y MODELO DE DESARROLLO.....	26
3.1.1 Lenguaje de programación.....	26
3.1.2 Técnicas de programación .....	27
3.1.3 Plataforma de desarrollo.. ..	28
3.1.4 Entorno gráfico.....	29
3.1.5 Lista de tareas.....	30
3.1.6 Ruta crítica .....	31
4 ANALISIS.....	32
4.1 REQUERIMIENTOS DEL USUARIO.....	32
4.2 DESCRIPCIÓN DETALLADA DE CADA PROCESO .....	33
4.2.1 Modelo-Vista-Controlador.....	36
5 DISEÑO.....	37
5.1 RECOLECCIÓN DE REQUERIMIENTOS.....	37
5.1.1 Módulos diseñados bajo el esquema Modelo-Vista-Controlador.....	39
5.2 VIABILIDAD DEL SISTEMA .....	40
5.2.1 Viabilidad técnica.....	40
5.2.2 Viabilidad operativa.....	40
6 MODELAMIENTO DEL SISTEMA .....	41
6.1 DICCIONARIO DE DATOS .....	41

6.2	CARPETAS .....	43
6.3	DIAGRAMA ENTIDAD-RELACIÓN .....	44
6.4	DIAGRAMA RELACIONAL .....	45
6.5	CASOS DE USO .....	45
6.6	DIAGRAMAS DE ESTADO .....	47
6.6.1	Diagrama de estado administrador:.....	47
6.6.2	Diagrama de estado Estudiante .....	47
6.6.3	Diagrama de estado profesor .....	48
6.7	DISEÑO DEL SISTEMA PROPUESTO.....	48
7	DESARROLLO .....	51
7.1	ESPECIFICACIÓN DEL SISTEMA.....	51
7.1.1	Destinatarios de la aplicación.....	51
7.2	ESPECIFICACIONES TÉCNICAS .....	51
7.2.1	Software.....	51
7.2.2	Hardware.....	52
7.3	ESTRUCTURA DEL CÓDIGO FUENTE .....	52
8	PRUEBAS.....	57
8.1	PRUEBA DE CAJA BLANCA .....	58
8.2	PRUEBA DE CAJA NEGRA .....	60
9	GLOSARIO .....	61
10	RESULTADOS OBTENIDOS.....	64
11	CONCLUSIONES .....	65
12	SUGERENCIAS Y RECOMENDACIONES .....	66
13	BIBLIOGRAFÍA .....	67
14	MANUALES .....	68
14.1	MANUAL DEL PROGRAMADOR.....	68
14.1.1	Listado de scripts.....	68
14.1.2	Listado de variables de sesión .....	70
14.2	Manual del usuario .....	71
14.2.1	Ingreso al sistema. ....	71
14.2.2	Usuario administrador. ....	72
14.2.3	Usuario profesor. ....	76
14.2.4	Usuario estudiante.....	79
14.2.5	Usuario sin registrar. ....	81

## Índice de tablas

Tabla 1: hitos ó lista de tareas.....	31
Tabla 2: Diccionario de datos.....	42

## Índice de gráficos

Grafico 1: Funcionalidad PHP.....	20
Grafico 2: Funcionalidad del esquema Modelo-Vista-Controlador.....	28
Gráfico 3: Ruta critica .....	31
Grafico 4: Diseño general de lógica para <i>el cuarto de lavado</i> .....	38
Grafico 5: Diseño interfaz modulo <i>escribir</i> .....	39
Grafico 7: diagrama entidad-relación .....	44
Grafico 8: diagrama relacional .....	45
Grafico 9: caso de uso de todos los usuarios del sistema .....	45
Grafico 10: Caso de uso <i>estudiante</i> .....	46
Grafico 11: Caso de uso <i>docente</i> .....	46
Grafico 12: Caso de uso <i>administrador</i> .....	46
Grafico 13: Diagrama de estado de <i>administrador</i> .....	47
Grafico 14: Diagrama de estado <i>estudiante</i> .....	47
Grafico 15: Diagrama de estado <i>profesor</i> .....	48
Grafico 17: Diseño de ventana principal <i>estudiante</i> .....	49
Grafico18: Diseño ventana principal docente .....	49
Grafico 19: diseño del módulo escribir.....	50
Grafico 18: diagrama de flujo <i>subir archivos</i> .....	58
Grafico 19: Diagrama de grafos <i>subir archivos</i> .....	59



# INTRODUCCION

Teniendo en cuenta la creciente cantidad de información en la web sobre diversos temas, en formato de texto y multimedia, y las barreras de espacio en los dispositivos de almacenamiento, cosa que repercute en la calidad de herramientas informativas y educativas como, por ejemplo, las enciclopedias, además, la capacidad de la red tanto en su bastísimo espacio disponible y alcance global de la información, hace del siber espacio, con las aplicaciones convenientes, un campo al que se puede dar un enfoque educativo y muy interactivo.

Adicionalmente la tecnología puede entrar perfectamente a prestarse como herramienta en la metodología educativa, o mejor aún, la disposición de una herramienta eficaz y sobre todo flexible modificará la metodología mejorando la calidad del aprendizaje pues se tendrán nuevas formas de representar la información además de espacio disponible “infinito”.

Pero ¿cuál es el objetivo de este planteamiento?, sencillamente se propone presentar una aplicación CMS (Content Management System) docente-estudiante, puesto que de las modalidades informáticas, es esta la que más puede adaptarse a las exigencias de cada usuario, dadas sus facilidades de expansión, uso y configuración. Esto hace posible la presentación de contenidos de forma digital e interactiva como no se puede encontrar en la cotidianidad académica. La gran ventaja será que la información no se limitará a la recopilación de información que haga el grupo de programadores para dársela al usuario, sino que estos últimos también podrán hacerlo, simplemente cumpliendo ciertos estándares al momento de crearla y agregarla.

El sistema no esta enfocado a presentar una temática especifica dentro del mismo, sino a ofrecer una plataforma donde cualquier tipo de tema pueda ser subido, consultado ó trabajado. Es decir, la temática central irá siendo guiada por los intereses de los usuarios que la compongan.

Los formatos soportados por el sistema para la presentación y explicación de los contenidos son los siguientes:

- 3D o animaciones de 2D en flash (swf)
- Fotografías (gif, mpg, png)
- Texto (txt)
- video (mpeg, wmv, avi, mov)
- audio (mp3)
- Y extensiones autorizadas por el administrador para el “upload” de información en diversos archivos.

## 2 TEMA

El tema del presente trabajo es: ¿Cómo ampliar la información suministrada en un salón de clase, de modo que sea asequible fuera de éste y de paso amoldable a los horarios de los estudiantes, adicionalmente lo suficientemente didáctica para hacerla más comprensible?, en el caso de que el estudiante mismo se encuentre interesado, claro.

Dentro de la preparación académica (tecnología en informática) puede deducirse ya la respuesta: se permite adaptar una solución informática a este problema específico. Hay que aclarar que dicho problema puede tener también salidas por diversos caminos diferentes, ajenos a lo que le concierne a los tecnólogos. Aparece entonces en el horizonte no solo la creación de una herramienta de software sino que en pro de aumentar el impacto, de abarcar mucho más, en realidad la solución que estamos describiendo, la aplicación de software debe estar dirigida a la web, de modo que se pueda acceder a lo que ella contiene de forma remota, en todo el sentido de la palabra.

En el momento histórico que vive esta generación, el grado de desarrollo tecnológico de la humanidad se encuentra bastante avanzado, en el sentido de que esta pluralidad de posibilidades no solo remplazará muchas de las funciones que hacían individuos sino que las superará, volviéndose entonces imprescindible. Todo esto se dirige a que la tecnología en su súper abundancia nos presenta, nos da a disposición, en nuestro campo, herramientas de software libre y cantidad casi ilimitada de información de cómo utilizarla. En caso específico de *“El cuarto de lavado”* la web por medio del HTML y PHP.

### 2.1 TÍTULO DEL PROYECTO

*“El cuarto de lavado. CMS power”.*

El proyecto adquirió dicho título por varias razones:

- Al gestionar éste cuentas de usuario donde se maneja mucha información del propio usuario y filtros que evitan mostrar o llegar a lugares no autorizados, viene entonces a la cabeza la palabra privacidad. Privacidad es sinónimo de intimidad y esta se vive en nuestro cuarto de descanso.
- Alusión a la soledad, pues se dice que en algunos tal cosa es productiva
- Lavado se toma como parte un proceso de filtración, donde se deja lo que sirve, donde se deja lo que somos.
- En honor a un artista admirable que usa esta frase en una de sus canciones.

## 2.2 DESCRIPCIÓN DEL PROBLEMA

Se tiene por un lado un docente y por el otro un grupo de estudiantes. Puede haber una falta de claridad en los temas que afecte el rendimiento académico, atrasando el cumplimiento del pensul y demás perjuicios en la preparación final del estudiante, debido a diversos factores como pueden ser falta de interés, de seguimiento del tema ó claridad por parte del docente.

Se conoce el poder de difusión que tiene la información colgada en la red. Y el uso de esta información esta siempre a disposición de cualquier individuo, sea que éste esté o no preparado para recibirla. Es similar a como ocurre en una biblioteca, solamente que aquí se encuentra muchísima más información, aún aquella que vive al margen de los valores imperantes.

Es un hecho también que la mayoría de la población estudiantil puede tener fácil acceso a la información vía web, sea en un local que provea el servicio ó en la misma vivienda, fenómeno al que se atiende para usarlo de forma progresista, didáctica y constructiva por medio del sistema que presentamos, donde se pone al servicio de la enseñanza académica la multimedia, la red y demás recursos actualmente existentes para la red.

Dicho lo anterior se tienen dos ingredientes que se pueden unir para ofrecer una alternativa a la población académica que ayude a dar luz en el camino de la educación. El docente prepara en forma digital el tema que en sus cátedras enseña y lo sube al sistema, el estudiante al “logearse” ve los temas presentados con un formato al estilo enciclopedia que su docente ha subido (pues estos están relacionados). Es posible también que el estudiante escriba artículos para que empiecen a ser consultados una vez subidos.

Un CMS es presentado aquí como la solución a la problemática descrita. Obviamente esto no sustituye el esfuerzo individual de ambos lados, pero si como una forma para que el profesor saque sus temas fuera del aula y de los horarios de clase y los ponga a disposición libre a sus estudiantes. ¿Esto que quiere decir? La aplicación tendrá a disposición una biblioteca multimedia, donde el audio, el video, la animación y las fotografías entre otros, complementarán las diferentes temáticas. El profesor entrará como administrador acomodando dichos objetos en una interfaz para que sea visualizada por el estudiante a través de la red.

## **2.3 JUSTIFICACIÓN**

Se tienen en cuenta dos factores importantes que motivan la creación de este sistema: Lo primero y más importante es, que se pretende mostrar al usuario un sistema educativo donde el contenido esté en función de las expectativas subjetivas del mismo usuario, es decir, que sea lo más didáctico posible para la interiorización de la información, debido a la presentación de la misma; texto, imágenes, videos, audio, etc. Así podrá la aplicación acomodarse a la forma en que cada uno aprende.

El segundo factor que motiva este estudio, es la constante queja de parte de los docentes por la “escasa” creatividad de los estudiantes, que siempre escogen como proyectos de grado los mismos proyectos desde hace unos años (bases de datos para manejo de negocios pequeños, medianos o grandes, generalmente), tal vez gracias a las limitadas herramientas para programar que ha enseñado la universidad a sus estudiantes o puede ser también la poca pasión e indisciplina de estos académicamente y que por esto escogen proyectos relativamente fáciles por salir del paso.

## **2.4 ANTECEDENTES**

En un principio se pensó hacer una enciclopedia temática, pero dado que dichas herramientas son variadas y asequibles incluso a bajos precios, se descartó de inmediato. Sin embargo esta era una alternativa para ofrecer un producto informático útil. Podría unirse entonces ciertas funciones a esta idea, aprovechando las posibilidades de las bases de datos y el lenguaje. Se pensó más tarde en manejo de usuarios con sus correspondientes cuentas donde el sistema se limite a la actividad educativa, es decir, relación estudiantes-docentes, creación de artículos, manejo de archivos, agrupación de miembros, envío de mensajes etc. Entonces son dos factores que se unen; la consulta pública de artículos y manejo de grupos (miembros) e información por medio de cuentas personales y base de datos. Se pensó en un sistema CMS que integrara de una vez todo.

## **2.5 DESCRIPCIÓN DEL SISTEMA ACTUAL**

En la actualidad este sistema no existe en la agrupación que “El cuarto de lavado” pretende. Es suficientemente conocido por todos los estudiantes, en este caso de la uniminuto, el sistema de aulas virtuales, donde los estudiantes para entrar (acceder a todo el material allí alojado) son inscritos por un profesor que a su vez es el director de una materia identificada con un “nrc”. Ésta

materia debe ser creada por un administrador del sistema que se encarga también de asignar el profesor a la materia y proveer passwords en cualquier momento. Básicamente el aula virtual es un espacio para el envío y recepción de mensajes (tareas para los estudiantes), se alojan archivos y se hacen exámenes.

*El cuarto de lavado* no es específicamente un sistema para medir al estudiante, es decir no es un espacio obligatorio. Significa que no se hacen allí exámenes, aunque si es posible contabilizar la frecuencia de uso por parte de un estudiante. El sistema pretende complementar una información dada en un aula de clase (cosa que también se puede hacer con las aulas virtuales) usando medios multimedia como video, audio en línea y los tradicionales texto, imagen y downloads de archivos. Adicionalmente se pueden crear aulas virtuales, las cuales se entienden de la siguiente manera:

- El profesor es agregado por el administrador
- El profesor puede crear a su antojo materias (aulas ó grupos de sus estudiantes).
- En dichas materias puede agrupar estudiantes quienes verán el material subido por él.
- Entre todos los miembros de un grupo se pueden enviar mensajes; a uno en particular o a todo el grupo.

El cuarto de lavado quiere ser también una enciclopedia on-line, donde los artículos subidos por sus usuarios sean consultados en su pagina principal (para usuarios sin registrar) y también dentro de cada una de las cuentas de los miembros (usuarios registrados).

Como se ve el proyecto pretende, si no superar a las aulas virtuales reemplazándolas, si por lo menos crear una comunidad donde por medio del sistema se les enfoque a actividades más productivas.

## **2.6 OBJETIVOS**

**2.6.1 Objetivo general.** Crear un software administrador de contenidos multimedia orientado a la web con el fin de convertirse en una herramienta educativa que complemente las metodologías llevadas por los docentes y presente información pública para la consulta externa.

### **2.6.2 Objetivos específicos**

- Facilitar la expansión, uso y diseño de la información para consulta con una interfaz grafica agradable y de fácil uso que disminuya la utilización de plataformas de gestión ajenas.
- Presentar al usuario por medio de herramientas orientadas a la web, temas que sean más amplios ó didácticamente mejor sustentados.
- Permitir el crecimiento de la aplicación por medio de adiconamiento de información en múltiples formatos, cuyos guías sean los usuarios del sistema, para acomodarse a las exigencias del momento.

## **2.7 METODOLOGÍA DE INVESTIGACIÓN**

El presente proyecto se esta realizando bajo el esquema de trabajo del desarrollo de prototipos, ya que la metodología llevada hasta ahora se caracteriza por empezar desde un desarrollo de los rasgos mas generales y en el camino se centra, se detiene en detalles que son fundamentales para el cumplimiento del objetivo final del sistema. Por lo mismo, se tiene una idea general, esto significa que no hay una completa desambigüedad, ni recolección precisa de los requerimientos para un riguroso proceso de análisis que dicte un camino especifico por donde se debe andar para llegar a un dicho resultado final definido, eso si, desde el principio. Dado el caso se preparan prototipos para así llegar a definir correctamente los requerimientos al ver una parte del aplicativo en ejecución. Se desarrolla por módulos dada la envergadura de procesos tan diferentes el uno del otro que se deben hacer.

Además dadas las condiciones de desarrollo del aplicativo y la creación del proyecto de grado, el modelo de creación de prototipos surge como el más adecuado por la relativa rapidez y porque siempre existe una parte del sistema ya en funcionamiento.

¿Pero brevemente en que consiste el modelo de creación de prototipos?

Llamado modelo de prototipos o modelo de desarrollo evolutivo, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado (en forma de un diseño rápido).

Ventajas:

- Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

**2.7.1 Línea de investigación.** En la Corporación universitaria Minuto de Dios se han establecido 4 líneas de investigación:

- Línea 1: Innovación educativa y transformación social.
- Línea 2: lenguaje, comunicación y pensamiento.
- *Línea 3: Innovaciones tecnológicas y cambio social.*
- Línea 4: Gestión, participación y desarrollo comunitario.

Teniendo en cuenta claramente los anteriores puntos, se puede inferir que el presente trabajo se encuentra suscrito a la línea número tres (3), al igual que la facultad de ingeniería. Para el desarrollo, cada numeral cuenta con sublíneas de investigación que rigen su orden y su objetivo. Para el programa Tecnología en Informática son:

- *Sublínea 1: Desarrollo de Software*
- Sublínea 2: Sistemas de Información
- Sublínea 3: Redes de Computadores
- Sublínea 4: Plataformas

Por el tipo de enfoque que tiene “*el cuarto de lavado*”, se adopta la sublínea de *Desarrollo de software* pues al ser este un CMS, permite visualizar cualquier tipo de información de consulta en formato enciclopedia, además de poder gestionar usuarios que a su vez pueden administrar sus propios contenidos y/o archivos, en pocas palabras manejar información por medio de cálculos prediseñados por los respectivos lenguajes de programación. Se puede decir también que el aplicativo es desarrollado utilizando varios lenguajes pues es una de las posibilidades del grupo de plataformas donde corre éste para el usuario final, es decir el diseño de dichas plataformas permite que varios lenguajes de programación se complementen.

## 2.8 ESTADO DEL ARTE

Uniendo la experiencia y la investigación, se han de nombrar en primer orden dos sistemas CMS cuyas funciones son muy semejantes a las del aplicativo aquí presentado. Son los sistemas “Joomla!” y “Mambo” diseñados bajo los esquemas de software libre. Igualmente las aulas virtuales de la Corporación Universitaria Minuto de God, comparte algunas funcionalidades con el sistema de este proyecto.

Joomla es un sistema gestor de contenidos dinámicos (CMS o Content Management System) que permite crear sitios web con una alta interactividad para el usuario, profesionalidad, diseño y eficiencia. La administración de **Joomla** está enteramente basada en la gestión online de contenidos. Está programado en lenguajes **PHP** (Hypertext Pre Processor), **SQL** (Structure Query Language), **HTML(Hyper Text Markup Language)**, **Java Script**, y **XML**. Al utilizar bases de datos relacionales, concretamente usa el sistema **MySQL**. El software Mysql es programa **OpenSource** de libre distribución y uso, precisamente esto es lo que ha creado comercialmente la “llave” entre MySQL y php, como consecuencia una compatibilidad que ofrece realmente estabilidad en todo el sentido de la palabra.

Por otro lado el sistema “Mambo” es la génesis de Joomla!. Esto significa que ambos sistemas son hermanos y sus características son semejantes; en cuanto a lenguaje, interfaz, métodos, soluciones, etc. Además existen en el mercado CMSs con enfoques diferentes, es decir administración de páginas web, de blogs, de fotoblogs, etc.

Por otro lado existe también un sistema llamado Moodle. Este es un sistema de gestión de cursos de libre distribución que ayuda a los educadores a crear comunidades de aprendizaje en línea. Se parte aquí la diferencia entre *el cuato de lavado* y *Moodle* pues el primero pretende ofrecer un uso más básico además de que se espera mejorar en posteriores versiones.

Desde luego con la incursión en el campo informático de lenguajes orientados a la web, poco a poco el mercado se ha llenado de más y mejores sistemas aparte de los mencionados arriba, pues la competencia y los alcances en la lógica así lo han permitido. A continuación se listan nombres de varios sistemas CMS parecidos los anteriores, cabe aclarar que estos están orientados a la administración de paginas web:

- Apache Lenya (Java/XML)
- ASP Nuke (ASP) Un CMS basado en ASP de código libre.
- CMS10 (PHP, SWF y Ajax) Gestor de contenidos de nueva generación.
- door108 (PHP) CMS totalmente en español basado en e107. Incluye novedades como sistema de administración MultiTarea y MultiArea para la creación de cientos de sitios sin ocupar casi espacio físico.



- DotNetNuke (.NET) CMS desarrollado en .NET, gratis y con fuentes. nota: Más que un CMS en sí es un Framework de .NET pensado para desarrollar CMS entre otras cosas.
- eZ Publish (PHP) CMS framework muy potente que sirve para páginas webs, intranets, comercio electrónico, extranets y portales.
- Gekko (PHP) CMS en español muy seguro, fácil de configurar y altamente escalable.
- Jaws (PHP) Framework y CMS amigable para el usuario y desarrollador.
- Jupiter Content Manager
- MemHT Portal (PHP)
- MODx PHP MODx es un derivado (Fork) de Etomite, resulta ser un CMS más versátil que otros demasiado estructurados.
- NukeET (PHP) CMS totalmente en español basado en el PHP-Nuke.
- Openflavor (PHP) Gestor de contenidos web en castellano.
- OpenCms (Java)
- Plone (Zope/Python) Muy flexible y poderoso, excelente interface de usuario, instalación muy limpia, buena cantidad de addons, impresionante grado de personalización, integración con LDAP u otros sistemas de login.
- PHP-Nuke (PHP)
- phpwcms (PHP) CMS orientado a la construcción de sitios web para profesionales y empresas.
- SPIP (PHP) Gestor de Contenido de licencia libre
- Textpattern (PHP)
- Tiki CMS
- TYPO3 (PHP) herramienta CMS con estructura multinivel, motor de búsquedas, gestión de autoría y publicación de contenidos, mecanismo de uso de plantillas para la maquetación de páginas, multilingüe,... Es también una herramienta portal: administra la personalización de las páginas según la identidad de los usuarios. Es enteramente extensible por módulos. Dispone de una comunidad muy activa.
- TYPOLight (PHP) potente CMS especializado en la accesibilidad. Utiliza XHTML y CSS para generar páginas que cumplen W3C/WAI. Desarrollado por Leo Feyer en 2004 bajo licencia GPL.
- WebGUI (Perl) Ocupa más de 40 MB, flexible, adaptable, multilingüe.
- Xaraya (PHP). Es un CMS bastante potente y general, aunque con una elevada curva de aprendizaje.
- XOOPS (PHP) CMS modular. Instalación sencilla, gran soporte comunitario, gran cantidad de módulos y themes, mucha funcionalidad, sistema de permisos muy bueno.
- Moodle: Plataforma educativa que permite evaluar a los miembros.

## 2.9 MARCO TEÓRICO

Es imprescindible el uso del lenguaje HTML, pues finalmente es este con el que se escribe la interfaz para el usuario. Sin embargo, principalmente se quiere utilizar el lenguaje PHP porque permite hacer operaciones complejas que de otro modo, incluso con Java Script no se podrían hacer, operaciones que pueden salirse del estricto enfoque de una página web, es decir con orientaciones más semejantes a las de una aplicación ejecutada sobre el escritorio. HTML no hace operaciones pues este se limita a darle al programador una variedad de herramientas de interfaz y formatear los atributos de dichos objetos. Adicionalmente PHP se creó para integrar fácilmente el lenguaje SQL pues ambos son de licencia libre además de que proveen soluciones a exigencias actuales, es decir manejo de grandes volúmenes de datos cosa que expande las vías y las fronteras de quien sepa explotárselas a su favor.

Quien piense en dar una interactividad, aumentar las funciones y posibilidades al usuario en una web casi al punto de hacerla parecer una aplicación, se encontrará con una respuesta: manejar, crear, consultar diversas variables sobre la sesión y volverlas a recuperar en posteriores utilidades. Implica entonces guardarlas en algún lado y consultarlas por parte de los que estén interesados. SQL es la solución y PHP la forma de acceder y utilizar la información allí alojada.

Se quiere que la aplicación funcione también para usuarios sin red, como si fuera una enciclopedia, esto se logra con un software que simule la máquina de red en la local porque (esta es otra razón más) PHP se ejecuta del lado del servidor para mostrar el resultado al usuario. Sin embargo se busca que se pueda utilizar sobre la red, pues es en realidad allí donde está el potencial, ya que es actualizada, manipulada, consultada remotamente. Se escogió también este lenguaje por su semejanza con C/C++, lenguaje que junto a JAVA son los que se enseñan con más ahínco dentro del plan del pensul.

Entonces ¿Cuáles son finalmente los lenguajes a utilizar y razones?

- PHP: Por su velocidad al ser ejecutado por el lado del servidor. Familiaridad. Se complementa bien con otros lenguajes (HTML, SQL) y es el indicado para llegar a cumplir los objetivos.
- HTML: Pues es inevitable al usar PHP. Familiaridad. Facilidad al tener herramientas que lo escriben automáticamente, lo cual reduce el tiempo tanto escribiéndolo como aprendiéndolo. Velocidad y diversidad de posibilidades para la interfaz.
- MySQL: Plataforma para manejo de información. Por compatibilidad con los otros lenguajes. Estabilidad, robustez y confiabilidad. Sencillo en su sintaxis. Comercial.

- JavaScript: Se recurrirá muy poco a este lenguaje pues el mismo PHP es una alternativa a éste. Sin embargo se puede encontrar en el camino con problemas que necesiten de una u otra sentencia Java Script para dar una mano, ó ejecutar una que otra acción.

Breve descripción de cada uno:

**2.9.1 PHP:** Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Características:

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.

- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de abierto, código PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.



grafico 1: Funcionalidad PHP

- Con PHP se puede hacer cualquier cosa que se logre hacer también un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

**2.9.2 HTML.** Es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Marcas Hipertextuales. Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores como Internet Explorer, Opera, Firefox, Netscape o Safari, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para web.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit, el Bloc de Notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad, Vim etc.

Existen además, otros programas para la realización de sitios Web o edición de código HTML, como por ejemplo Microsoft FrontPage, el cual tiene un formato básico parecido al resto de los programas de Office. También existe el famoso software de Macromedia llamado Dreamweaver, siendo uno de los más utilizados en el ámbito de diseño y programación Web. Estos programas se les conoce como editores WYSIWYG o What You See Is What You Get (en español: “lo que ves es lo que obtienes”). Esto significa que son editores en los cuales se ve el resultado de lo que se está editando en tiempo real a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios web, sino que una forma un tanto más simple ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML la cual va generando todo el código a medida que se va trabajando.

**2.9.3 SQL.** Lenguaje de Consulta Estructurado (Structured Query Language), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla. Es un lenguaje de cuarta generación (4GL)<sup>1</sup>.

---

<sup>1</sup> 4GL o lenguaje de cuarta generación son los lenguajes en los cuales en lugar de escribir cómo deben obtenerse los resultados se especifica qué resultados son los que se quiere obtener. Por ejemplo los lenguajes de consulta de base de datos (como el SQL) son considerados lenguajes de cuarta generación.

SQL explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos. Es un lenguaje declarativo de alto nivel o de no procedimiento, que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación. De esta forma una sola sentencia puede equivaler a uno o más programas que utilizaran un lenguaje de bajo nivel orientado a registro.

**2.9.4 JavaScript.** Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.

JavaScript es un lenguaje sin acceso al servidor, sin embargo la empresa Microsoft introdujo un desarrollo llamado XMLHttpRequest en el objeto Active X para IE 5.0 del año 2006, el cual combinando JavaScript y XML puede hacer peticiones al servidor, lo cual actualmente se ha convertido en Ajax. El presente proyecto utiliza librerías xajax la cual es la evolución de Ajax junto a php y da facilidad para convertir funciones php en JavaScript. Es sumamente utilizado dentro del modelo-vista-controlador y sobre todo en los scripts del control pues accede a funciones php escritas en scripts del modelo.

**2.9.5 Software libre.** En inglés *free software* se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades (libertad 1).
- La libertad de distribuir copias, con lo que puede ayudar a otros (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3).

Para la libertad 1 y 3 el acceso al código fuente es un requisito previo.

El software libre suele estar disponible gratuitamente (pero no hay que asociar software libre a software gratuito), o a precio del costo de la distribución de éste, sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.

Análogamente, el *software gratis* o *gratuito* (denominado usualmente freeware) incluye en algunas ocasiones el código fuente; sin embargo, este tipo de software *no es libre* en el mismo sentido que el *software libre*, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

No debe confundirse "software libre" con software de dominio público. Éste último es aquél por el que no es necesario solicitar ninguna licencia y cuyos derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público. En resumen, el software de dominio público es la pura definición de la libertad de usufructo de una propiedad intelectual que tiene la humanidad porque así lo ha decidido su autor o la ley tras un plazo contado desde la muerte de éste, habitualmente 70 años<sup>2</sup>.

## 2.9.6 Formatos de archivo a utilizar

**2.9.6.1 ¿Qué es FLV?** FLV es un formato de archivo de vídeo con compatibilidades avanzadas. Este formato es usado para transmitir vídeos sobre Internet. FLV requiere Flash Player de la versión 6 en adelante para ser instalado en el navegador de nuestro cliente.

El formato de audio en los archivos FLV regularmente se encuentra codificado como MP3. también soportan audio sin comprimir o en formato ADPCM (Adaptive Differential Pulse Code Modulation).

Los mismos son códecs de ondas que en vez de cuantificar la señal directamente, como códecs PCM (Pulse-Code Modulation), cuantifican la diferencia entre la señal y una predicción hecha a partir de la señal, por lo que se trata de una codificación diferencial.

- ¿Por qué elegir FLV?

Los archivos FLV contienen bit streams de vídeo los cuales son una variante del estándar H.264. FLV soporta dos nuevas versiones del códec "sceenshare", es un formato de codificación diseñado para screencasts. Estos formatos están basados en mapas de bits y pueden tener pérdida al reducir la profundidad de color y están comprimidos usando zlib.

---

<sup>2</sup>

Fuente [www.wikipedia.org](http://www.wikipedia.org)

Si utilizamos otros formatos de vídeos es probable que el usuario no cuente con un reproductor adecuado para ello, si aparte de eso utilizamos un códec de compresión y el usuario no lo tenga instalado en su ordenador no podrían observar el vídeo, a diferencia de FLV que requiere Flash Player.

- ¿Qué es H.264?

H.264 también conocido como MPEG-4 AVC, es un estándar que define un códec de vídeo de alta compresión, desarrollada conjuntamente con la ITU-T (ITU Telecommunication Standardization Sector), VCEG (Video Coding Experts Group) y el MPEG (ISO/IEC Moving Picture Experts Group).

La intención del proyecto H.264/AVC fue la de crear un estándar capaz de proporcionar una buena calidad de imagen con tasas binarias notablemente inferiores a los estándares anteriores (MPEG-2, H.263 o MPEG-4 parte 2), además de no incrementar la complejidad de su diseño.

- ¿Qué es screencasts?

Screencasts es una grabación digital de salida por pantalla de computadora, puede contener narración de audio. Screencasts son útiles para demostrar características de software. Ayuda a desarrolladores de software a mostrar su trabajo.

- ¿Qué es un códec?

Los códec de video son programas que nos permiten comprimir y descomprimir los archivos de vídeos digitales. Los algoritmos de compresión utilizados pueden llevar a cabo a pérdidas de información, lo cual podría debilitar la calidad de los vídeos.

Las iniciales de códec vienen dadas por las palabras COmpresor / DECompresor. La función de los códec de vídeo es reducir el tamaño de la información de los vídeos. Podemos observar por ejemplo durante la edición de un vídeo como su tamaño aumenta considerablemente aunque tenga una duración de pocos minutos.

Para ello fueron necesarios realizar estos algoritmos de compresión en tiempo real llamados códec. Estos códec de vídeos conllevan a pérdidas en la calidad, por lo que se deben utilizar códec que permitan una buena compresión con mínimas pérdidas de calidad.

**2.9.6.2 JPG.** (Joint Photographic Experts Group) es un algoritmo de compresión con pérdida. Esto significa que al descomprimir la imagen no obtenemos exactamente la misma imagen que teníamos antes de la compresión. Es el formato más utilizado para almacenar y transmitir archivos de fotos en la Web.

El algoritmo de compresión JPEG se basa en dos defectos visuales del ojo humano, uno es el hecho de que es mucho más sensible al cambio en la luminancia que en la crominancia, es decir, notamos más claramente los cambios de brillo que de color. El otro es que notamos con más facilidad pequeños cambios de brillo en zonas homogéneas que en zonas donde la variación es grande, por ejemplo en los bordes de los cuerpos de los objetos.

**2.9.6.3 GIF.** GIF es un formato sin pérdida de calidad para imágenes con hasta 256 colores, limitados por una paleta restringida a este número de colores. Por ese motivo, con imágenes con más de 256 colores (profundidad de color superior a 8 bits), la imagen debe adaptarse reduciendo sus colores, produciendo la consecuente pérdida de calidad.

Permite transparencia de 1 bit, de tal forma que cada pixel de la imagen puede ser o no transparente

**2.9.6.4 MP3.** MPEG-1 Audio Layer 3, más conocido como MP3, conocido también por su grafía emepetrés, es un formato de audio digital comprimido con pérdida desarrollado por el Moving Picture Experts Group (MPEG) para formar parte de la versión 1 (y posteriormente ampliado en la versión 2) del formato de vídeo MPEG. Su nombre es el acrónimo de MPEG-1 Audio Layer 3.

Después de este breve repaso por cada uno de estos lenguajes, debemos adentrarnos entonces a explicar las herramientas que hacen posible el funcionamiento, interpretación ó compilación para que dichos lenguajes cumplan su función.

Son básicamente tres los programas que deben estar en ejecución, al menos en el momento de compilación del aplicativo, para poder trabajar: PHP, Apache y MySQL. La red trabaja principalmente enviando información con hipertexto donde los marcadores (<...>) dan formato al texto. Los navegadores precisamente existen para interpretar este código por lo que no necesitamos de software adicional para poderlo ejecutar, salvo un browser.

PHP no es soportado por ningún navegador, es decir no lo interpreta, no lo entiende, es como hablarle en chino. Por eso necesita de un motor adicional que complementa al navegador por medio de una maquina local. PHP contiene las librerías y el motor de compilación que hace posible la interpretación del código.

Al ser PHP un lenguaje del lado servidor, necesita que la maquina local (donde están alojadas las paginas) que lo ejecute. Apache provee este servicio, dando una dirección IP local (127.0.0.1 generalmente), donde toda información enviada por este puerto será procesado por él, simulando un host.



MySQL Es el aplicativo de manejo de información, este es el motor y gestor de los archivos donde almacena los registros (base de datos). Estas herramientas deben configurarse para que Apache acceda a ellas según los comandos que llegan de la IP local así lo exija.

Actualmente y pensando en liberar a los desarrolladores de la engorrosa tarea de configurar estos tres elementos básicos, la comunidad GPL pone a disposición del publico diversos aplicativos que no solo incluyen estos y mas elementos, sino que automáticamente los configura, como XAMPP, PHP-Nuke, AppServ o EasyPhp.

## 3 INGENIERÍA DEL PROYECTO

### 3.1 METODOLOGÍA Y MODELO DE DESARROLLO

**3.1.1 Lenguaje de programación.** El lenguaje utilizado para la programación de *El cuarto de lavado* es PHP. Este lenguaje tiene una sintaxis muy parecida a c/c++, además de que comparte ciertas funcionalidades de éste pero orientado a la web, lo que da la posibilidad de hacer scripts con una complejidad lógica al nivel de los lenguajes orientados a aplicaciones de escritorio, integrando técnicas de programación como la orientada a objetos. Hay en la red también sin número de librerías disponibles escritas por comunidades, que pueden extender el potencial de PHP, como por ejemplo aquellas que hacen migración de información a formatos de documentos como pdf o Excel, ó para utilizar el motor de plantillas Smarty.

Entre los fuertes de PHP esta su capacidad para el manejo de bases de datos, pudiendo conectarse con Mysql, PostgreSQL, Oracle, ODBC, IBM DB2, Micosoft SQL Server SQLite. PHP Recibe datos desde diferentes lenguajes como HTML, JavaScript o Flash y la posibilidad de que sus aplicaciones corran en los sistemas operativos más populares como Linux, Windows y MacOS.

Adicionalmente se complementarán los scripts con JavaScript y pequeñas aplicaciones Flash, sobretodo para la labor de reproducción multimedia on-line.

JavaScript tiene la propiedad de ejecutar operaciones (funciones en la cabecera de las paginas) sin necesidad de refrescar pues es un lenguaje interpretado -por la estación de trabajo-, es decir, que no requiere compilación, lo que es muy útil para validar campos de formularios, abrir ventanas pop-up y dentro de sus alcances más altos -usados en el presente trabajo- las funciones sajas que refrescan solamente el pedazo solicitado más no la pagina entera.

El uso de Flash se limitará a flash player, es decir para uso de reproducción de flv y mp3 on-line desde el navegador, esto se hace por medio de un almacenamiento temporal de fracciones consecutivas de dichos archivos en una memoria temporal llamada buffer y con el fin de dar una reproducción continua. Los archivos flash son de extensión swf y contiene un lenguaje interno llamado action script, que no se tocara.

## 3.1.2 Técnicas de programación

**3.1.2.1 Estructurada:** Es muy común entre estudiantes llamar a este estilo programación en lombriz, pues el programa cumple su función empezando desde la primera línea y terminando con la última sin reutilizar ninguna parte del mismo. En primera medida es una de las consecuencias de los primeros pasos por el lenguaje php pues éste integra la interfaz html con porciones de scripts en cualquier punto de las etiquetas según sea la exigencia. Por supuesto esto está contemplado entre las malas prácticas de programación porque el programa final se hace más pesado por el gran número de líneas, más difícil de mantener y de actualizar, pues al existir mayor cohesión (mayor dependencia entre módulos y menor cantidad de éstos) y al hacer cambios podría sacarse de funcionamiento el programa completo, además está más propenso a errores por parte de la máquina y mayor lentitud en el desarrollo. Sin embargo existe la ventaja que permite personalizar cada una de las partes.

Es el paso anterior a la programación orientada a objetos. Esta programación permite la reutilización del código y facilidad en la actualización y mantenimiento, pues porciones lógicas que tienen una entrada un proceso y una salida se separan del código principal, sobre todo si dicho proceso necesita hacerse varias veces en diferentes partes del programa. El programador notará al avanzar en experiencia que la tendencia es no tener un código principal sino una función principal que dispara un complejo sistema de funciones.

**3.1.2.2 Orientado a objetos:** Se diseñó como forma que la aplicación final optimizara memoria en tiempo de ejecución, separación modular del código, reutilización y actualización del mismo buscando una baja cohesión de forma que los módulos sean más independientes unos de otros, concluyendo en estabilidad del aplicativo desarrollado. En el aspecto de reutilización de código se avanzó muchísimo con la propiedad de herencia que tienen las clases.

**3.1.2.3 Framework:** Es el conjunto de herramientas prediseñadas separadas en librerías escritas en uno o más lenguajes para el desarrollo de más aplicativos basados en sus posibilidades. Con las propiedades de la programación orientada a objetos, en la actualidad los frameworks permiten la separación entre el diseño y la lógica del programa. En el presente proyecto utilizaremos un framework que contiene librerías ajax y Smarty. Esto desemboca en una nueva tendencia de programación llamada modelo-vista-controlador.

**3.1.2.4 Modelo-Vista-Controlador (MVC):** Es la separación entre la interfaz, la interacción del usuario con la lógica por medio de eventos y el proceso lógico que es disparado por un evento. Es generalmente usado para desarrollos web.

**Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo o la gestión con una base de datos.

**Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario (HTML, PHP).

**Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

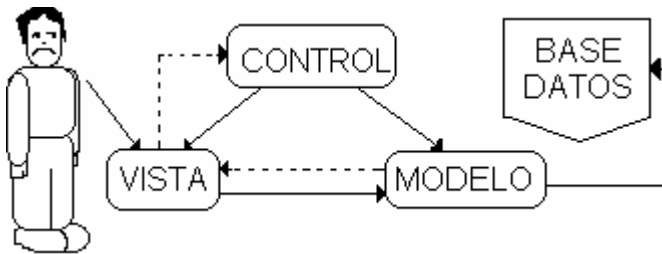


Grafico 2: Funcionalidad del esquema Modelo-Vista-Controlador

**3.1.3 Plataforma de desarrollo.** En primer lugar cabe aclarar que una plataforma de desarrollo es el lugar hardware o software sobre el cual se ejecuta un programa. Básicamente la plataforma de *El cuarto de lavado* es el servidor Apache, pues es sobre él donde toda aplicación web corre.

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 1 y la noción de sitio virtual, esto significa que Apache no solo se limita a ser servidor local pues desde sus inicios se baso en proporcionar una maquina servidor para la red. Hoy en día es utilizado por el 48% del hosting en el mundo<sup>3</sup>.

Según Wikipedia; Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Es necesario tener un compilador php para poder interpretar las paginas enviadas por Apache escritas en este lenguaje, y como utilizamos bases de datos necesitamos la plataforma mysql. Estos dos elementos necesitan ser

<sup>3</sup> <http://news.netcraft.com/>: cuentan con graficas de estudios estadísticos sobre la utilización y tecnologías en la web.

configurados para que sean utilizados por Apache según sea necesario, sin embargo se ofrecen paquetes que vienen con estas tres partes ya configuradas y solo basta instalarlas y empezar a camellar.

Para dicho trabajo utilice el ya muy popular EasyPhp 1.8.0.1, paquete que viene con lo siguiente, que se auto configura previamente:

- Apache 1.3
- PHP 4.3.10
- PhpMyAdmin 2.6.1
- MySQL 4.1.9

PhpMyAdmin es una aplicación escrita en lenguaje PHP para dar interfaz grafica a MySQL. De no ser así, nos dispondríamos a ejecutarlo y utilizarlo bajo símbolo del sistema.

Sin embargo en el transcurso del desarrollo del proyecto se tubo que migrar a un paquete similar al utilizado, ya que contaba con versiones más actualizadas de dichas plataformas. Esto fue impulsado por tres razones: lo exigió ciertos módulos escritos en nuevos métodos de programación; por eficiencia al asegurar que la aplicación pueda correr por estar diseñada en las últimas plataformas y porque no existía una versión nueva estable de EasyPhp. XAMPP para Windows (1.6.4) es el paquete utilizado, desarrollado por Apache Friends; consta de:

- Apache HTTPD 2.2.6 + Openssl 0.9.8e
- MySQL 5.0.45
- PHP 5.2.4
- PHP 4.4.7
- phpMyAdmin 2.11.1
- Zend Optimizer 3.3.0

**3.1.4 Entorno grafico.** PHP permite mezclarse con HTML en cualquier punto del código, por lo que el entorno grafico esta sujeto a toda la potencialidad de éste y lo soportado por el navegador. Puntualmente el aplicativo utiliza el lenguaje de etiquetas HTML para hacer la interfaz grafica, complementado con javaScript y plugins flash para la presentación de videos y música desde el navegador. No se utilizaran hojas de estilo de cascada CSS aunque es una de las poderosas facilidades del lenguaje para personalizar la interfaz con el usuario.

### 3.1.5 Lista de tareas

<b>N. T.</b>	<b>Lista de tareas</b>	<b>Tiempo</b>	<b>Dependencia</b>
T1	Identificación de necesidades	10 días	
T2	Viabilidad técnica y económica	8 días	T1 (M1)
<b>ANÁLISIS</b>			
T3	Estado del arte	15 días	T1, T2 (M2)
T4	Recolección de requerimientos	10 días	T1, T2 (M2)
T5	Documentación	20 días	T3, T4 (M3)
T6	Análisis Módulos	25 días	T5 (M4)
T7	Análisis Base de Datos	10 días	T6 (M5)
<b>DISEÑO</b>			
T8	Diseño modelo entidad-relación	4 días	T7 (M6)
T9	Diagramas UML	20 días	T8 (M7)
T10	Diseño interfaces	25 días	T9 (M8)
<b>CODIFICACIÓN</b>			
T11	Construcción base de datos	3 días	T8 (M9)
T12	Construcción módulo usuario sin registrar	7 días	T10, T11 (M10)
T13	Construcción módulo Administrador	20 días	T10, T11 (M10)
T14	Construcción módulo Estudiante	25 días	T10, T11 (M10)
T15	Construcción módulo Profesor	27 días	T10, T11 (M10)
T16	Construcción módulo Creación Artículos	2 días	T10, T11 (M10)
T17	Construcción módulo Buscar	3 días	T10, T11 (M10)
T18	Construcción módulo Subir Archivos	5 días	T10, T11 (M10)
T19	Construcción Interfaz	25 días	T10, T11 (M10)
<b>IMPLEMENTACIÓN</b>			
T20	Pruebas	15 días	T11, T12, T13, T14, T15, T16, T17, T18, T19 (M11)
T21	Manuales	15 días	T21 (M12)

Tabla 1: hitos

### 3.1.6 Ruta critica

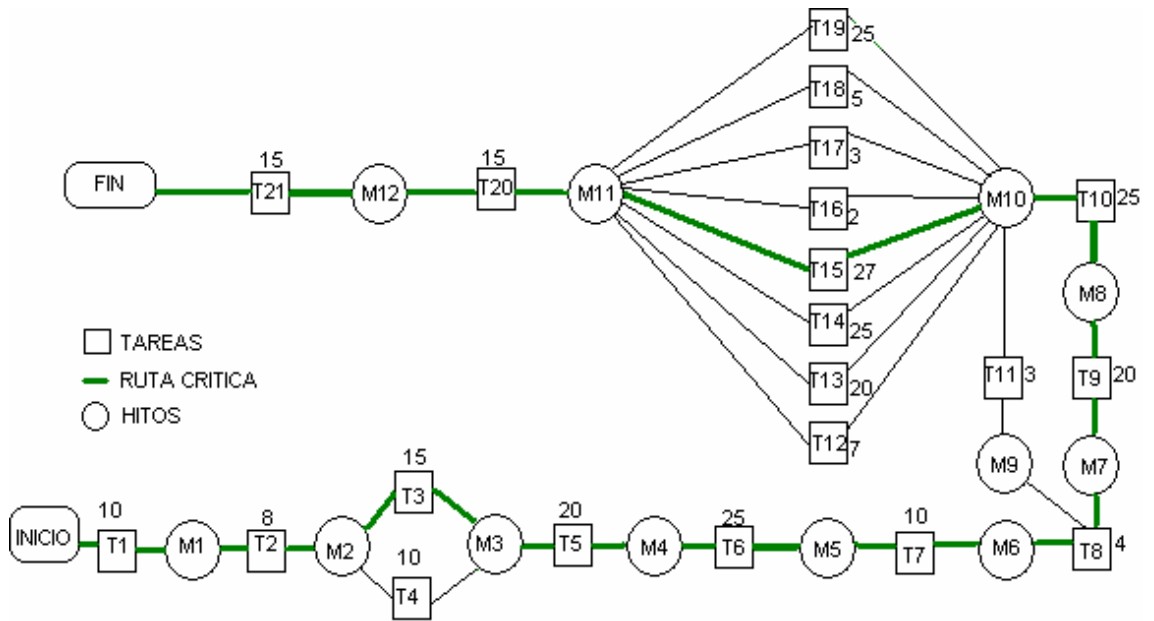


Gráfico 3: Ruta critica

Ruta critica: T1, T2, T3, T5, T6, T7, T8, T9, T10, T15, T20, T21= 194 días

## 4 ANALISIS

### 4.1 REQUERIMIENTOS DEL USUARIO

- Se quiere que el sistema tenga:
  - Manejo de archivos (subir, ejecutar, bajar, consultar, ver propiedades, tener dueño, validar tipo de archivos).
  - Los usuarios serán de cuatro tipos diferentes: estudiantes, docentes, no registrados y un administrador.
  - Maneje cuentas de usuario (Sesiones, Almacenar info de usuarios).
  - Gestor de texto (en caso que no se tenga un archivo a subir para escribir artículos).
  - Buscador de artículos y archivos, para externos y miembros (consultar artículos).
  - Creación de grupos (aulas), donde el docente personalice y tenga un grupo objetivo para sus temas.
  - Plantillas (interfaz).
  - Constara de dos tipos de interfaz, pues habrá un usuario administrador y otro visitante.
  - Creación de un gestor administrador. Tendrá privilegios totales, podrá ver, agregar, eliminar y editar contraseñas, miembros, archivos, temas sin ningún tipo de limitaciones.
  - Creación de base de datos, donde estarán las direcciones y propiedades de archivos alojados en el host (carpetas).
  - Mensajería entre miembros de grupos creados por los docentes.



## 4.2 DESCRIPCIÓN DETALLADA DE CADA PROCESO

- Pagina index: Esta pagina contiene los siguientes componentes: buscador, multimedia rand, links de información al cliente y los campos tradicionales para loguearse. Esta pagina muestra artículos, video y audio. Esta disponible para el tipo de usuario no registrado, por lo que desde ésta se tiene acceso a toda información que los usuarios registrados hayan introducido con la propiedad de "publico".
- *Buscador*: Esta es una sección presente en tres instancias diferentes: Index, Estudiantes, Profesores. La interfaz pone a disposición del usuario dos campos; el primero un campo de texto para entrar la palabra clave y el segundo un menú desplegable que cumple las veces de filtro (audio, video, imagen, artículo). Esta información es llevada a un modulo que busca en la base de datos según los criterios y devuelve un array con los correspondientes "id"s de los campos en las tablas donde fue encontrado un "match". El módulo esta diseñado para que no se repitan "ids". Otro modulo hace complemento al *buscador* al hacer las consultas correspondientes con el array devuelto y muestra un resumen en la interfaz para el usuario. A su vez a dicha información se puede acceder porque tiene un link relacionado.
- *Multimedia rand*: Este es un modulo que solo aparece en la pagina index. Allí se muestran videos cargados en el sistema (que han sido subidos por algún usuario, con privilegios "publico") aleatoriamente. De la misma manera funciona el audio. Ambos módulos reproducen el material de forma on-line, es decir desde el navegador, esto se logra mediante la utilización de plugins hechos en flash. El plugin de video utiliza un archivo XML de donde extrae la ruta de los archivos multimedia (flv y mp3) para reproducirlos. Utilizando funciones de archivos *El cuarto de lavado* reescribe dicho fichero con información de la base de datos de forma que el video no sea siempre el mismo. El plugin de audio funciona de la misma manera.
- *Campos de "login"*: Este consta de cinco elementos visibles al usuario: nombre, password, enviar, registrarse y recordación de password. Al enviar la información es recibida por un modulo que valida los campos, es decir rectifica que no estén vacíos y que la información sea la correcta, luego inicia variables de sesión que necesita el sistema y direcciona al usuario a su sesión correspondiente dependiendo del tipo de usuario que sea (administrador, profesor, estudiante).
- *"Recordación de password"* es un link que lleva a un modulo especial. Allí se le pide al usuario información solicitada en el momento de registro: Nombre, nit (cedula), pregunta (el administrador la predefine) y respuesta (string dado

por el usuario). Al ser enviada se validan los campos, luego si toda la información es correcta muestra el nick y el password en pantalla.

- En la opción registrarse el usuario por si solo, puede llegar al formulario de inscripción de estudiante. Para inscribirse como otro tipo de usuario (profesor) necesita el aval del administrador. El formulario pide las siguientes campos:

*Nombre:* Nombre del estudiante.

*Nick:* Apodo, para entrar en el formulario del index.

*Cedula:* Numero de identificación único, preferiblemente cedula.

*Mail:* Correo electrónico.

*Password:* Clave personal.

*Confirmar password:* Rectificar la clave que se acaba de dar.

*Pregunta:* Predefinida por el administrador. Para caso de olvido de password. Rectificar la clave que se acaba de dar.

*Respuesta:* A la pregunta anterior.

Administrador: Se diseño este modulo con el fin de un usuario tenga la posibilidad de configurar el sistema, ver, agregar, editar, eliminar. Las opciones que aparecen al entrar son las siguientes:

- *Registrar profesor:* esta opción lleva al modulo de registro del profesor donde dicho formulario tiene los campos de:

*Nombre:* Nombre del profesor.

*Materia:* Materia que dicta el profesor.

*Nick:* Apodo, para entrar en el formulario del index

*Cedula:* Numero de identificación único, preferiblemente cedula.

*Mail:* Correo electrónico.

*Password:* Clave personal.

*Confirmar password:* Rectificar la clave que se acaba de dar.

*Pregunta:* Predefinida por el administrador. Para caso de olvido de password

*Respuesta:* A la pregunta anterior.

*Registrar estudiante:* Los campos son los mismos descritos numerales arriba.

- *Nuevo artículo (U, P, Ad):* Este módulo tiene dos modalidades; un modulo para subir archivos al servidor y otro para escribir texto y poner imágenes. Al subir archivos con las extensión txt se puede ahorra el tiempo de escribirlo en el editor de texto para almacenarla en la base de datos. ¿Qué quiere decir? Que los archivos txt podrán ser visualizados como si el usuario hubiese escrito el texto desde el editor, a diferencia del resto de formatos de archivo (doc, pdf) que deben ser descargados.

- *Todos los archivos (U, P, Ad):* En esta opción despliega una consulta que muestra todos los archivos subidos por el respectivo usuario, es decir sus

archivos. Para el administrador se muestran todos los archivos subidos al sistema. Aquí se pueden eliminar y para los usuarios se pueden editar aquellos que han sido escritos desde el editor.

- *Usuarios (Ad)*: Se despliega una consulta con todos los usuarios que hacen parte del sistema, divididos entre profesores y estudiantes. Aquí se pueden eliminar o editar.
- *Configuración (Ad)*: Una opción interesante pues es justo aquí donde se pueden hacer ajustes importantes del sistema, como lo es la gestión en la conexión con la base de datos, los archivos que se pueden subir y etc. *El cuarto de lavado* tiene un módulo llamado “configuración.php” donde almacena constantes para dichos fines, éstas son manipuladas por el modulo “configuración” del administrador, a continuación se muestra código de definición de variables:

```
$r_a = 'http://localhost/kurt/'; ruta absoluta del host ($_SERVER['server_name'])
$host = 'localhost';
$usuario= 'root';
$password = "";
$base_datos = 'lavado';
$extensiones= 'txt,doc,gif,png,mp3,mp4,jpg,wmv,swf,flv';
$ruta= 'userfiles/';
```

Adicionalmente aquí se agregan las preguntas predefinidas que saldrán en el formulario a la hora que un usuario nuevo se registre.

- Estudiantes: Este es el espacio personal del usuario. En primera instancia tiene la opción de cambiar su imagen. En segunda medida esta informado siempre de que profesor lo ha agregado a su curso (pueden aparecer varios), dicho profesor tiene un link para consultar sus últimos movimientos respecto a sus artículos nuevos. Se tienen las siguientes opciones:
- *Subir*: Se dirige el modulo que permite subir archivos. Es dicho modulo cada uno de los archivos se identifican en, general, con un nombre, una materia y unas palabras claves de forma que puedan ser consultadas de manera mas precisa por el buscador. Este contiene un script que valida todo archivo subido; si la extensión es permitida por el administrador y si no esta repetido. Al ser esta validación verdadera el script renombra el archivo de manera que se sepa de quien es y cuyo nombre no se repita nunca; el proceso es el siguiente: el nombre se conserva pero al principio del string del nombre se le agrega una variable “unico”: NitUsuario\_Fecha(ddmmyyy\_H\_)\_nombreArchivo.nnn “H” es el valor de la hora actual sin minutos ni segundos, por lo que el mismo archivo puede ser subido de nuevo por el mismo usuario en la hora siguiente.
- *Mensajes*: Modulo con todo el manejo necesario de mensajería interna dentro de los usuarios que hagan parte de un aula: muestra si hay mensajes nuevos y un area de texto para enviar mensajes a los compañeros. El campo “para” contiene información sacada de una consulta donde aparecen los

compañeros que hacen parte de un aula específica (recordemos que las aulas con sus usuarios solo pueden ser creadas por los profesores). Se tiene disponible la opción de “boletín” que es enviar un mismo mensaje a todos los miembros del aula.

- *Salir*: Este es una opción que simplemente destruye la sesión y direcciona al usuario a la página principal. Esta hecha más precisamente por seguridad.

Profesores: Este es el espacio privado del docente, por eso puede modificar su imagen. Tiene exactamente las mismas opciones que el módulo de estudiantes. Sin embargo cuenta con otras opciones diferentes que marca la diferencia con respecto a la funcionalidad. El docente debe crear sus grupos de estudiantes los cuales tendrán acceso a mensajería entre ellos y fácil llegada a sus artículos.

- *Aula*: Este módulo tiene las siguientes opciones de manejo; Al pulsar sobre “aula” se muestran dos opciones; editar una ya existente ó crear una nueva. Al crear una nueva el sistema le pide una materia para el aula, deberá ser escrita. El script relaciona el nombre entrado con el nit del usuario que la creó, por lo cual un mismo nombre puede ser usado por muchos profesores sin peligro que se crucen los cables y los estudiantes que hacen parte de dicho grupo. Luego el profesor puede buscar a sus estudiantes entrando la correspondiente cedula o en una consulta que le muestre todos los estudiantes registrados del sistema. Acto seguido podrá inscribirlos uno a uno. No es posible repetirlos en una misma aula pues el script no lo permite. Los estudiantes tienen fácil acceso a los artículos escritos por el profesor además de mensajería interna entre ellos.

#### **4.2.1 Modelo-Vista-Controlador**

*Escribir artículo*: Este módulo está escrito bajo la técnica MVC. No permite subir archivos. Aquí se escribe el artículo en un editor de texto web (FCKeditor) y se relaciona con el mismo método de subida de archivos (nombre, materia, palabras claves).

## 5 DISEÑO

### 5.1 RECOLECCIÓN DE REQUERIMIENTOS

Nos encontramos entonces con las siguientes necesidades:

- Es necesario tener una pagina "index" donde esté el formulario básico para "loguearse" y entrar al sistema. Es impredecible que el primer pantallazo sea muy llamativo, pues esta pensado también para el usuario de paso que no esta interesado en hacer parte del sistema.
- Dicho index debe tener acceso a archivos cargados (buscador) por usuarios miembros, que durante el proceso de subirla hallan decidido darle privilegios públicos.
- Debe recuperar password.
- El sistema debe crear aulas o grupos donde se puedan ver mensajes y archivos entre los usuarios.
- Es necesario trabajar con variables de sesión (en lenguaje php, esto es posible), ya que esto evita la utilización de formularios "hidden" revelado información relevante.
- Cada pantalla que aparezca después de un login debe pasar por un proceso de validación antes de ser revelada su información. Esto como medida de seguridad.

El siguiente grafico es a grandes rasgos una explicación del funcionamiento del sistema *El cuarto de lavado*, nótese que el último pantallazo donde aparece finalmente la carreta hace parte de los módulos “estudiante”, “profesor” y “no registrado” ó pagina publica ”index”, es decir, estos usuarios pueden investigar y desplegar artículos:

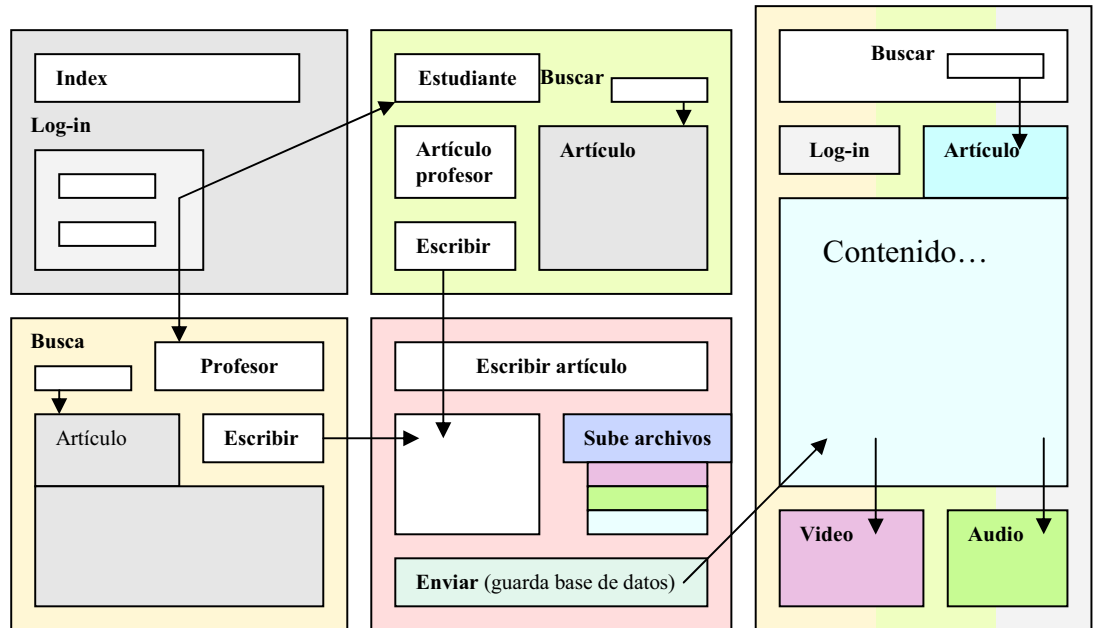
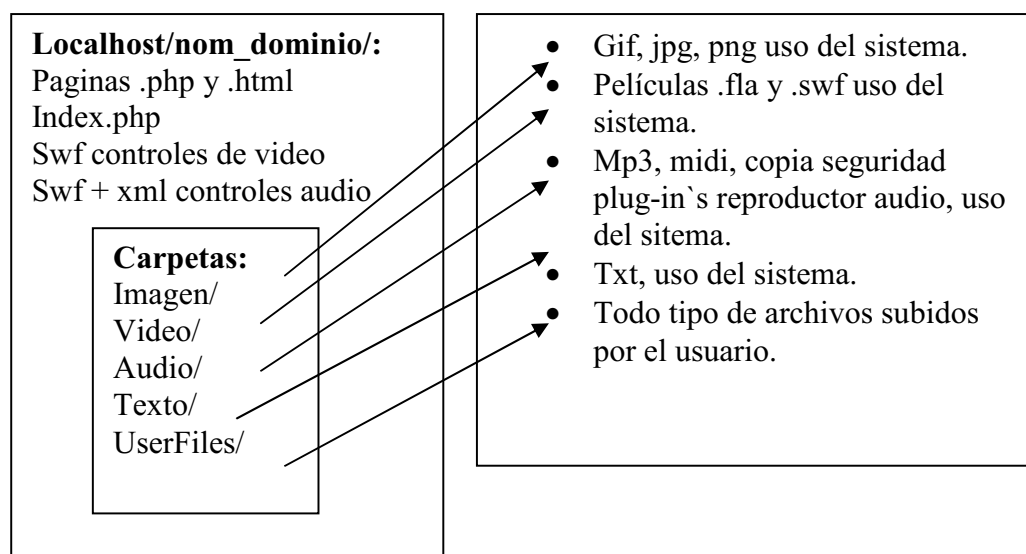


Grafico 4: Diseño general de lógica para *el cuarto de lavado*

La organización de los módulos del CMS en disco:

Host:



### 5.1.1 Módulos diseñados bajo el esquema Modelo-Vista-Controlador

Así se hace la distribución del módulo *escribir* (el cual esta diseñado bajo el esquema MVC), donde se utiliza la plataforma *Smarty* pues ésta hace posible separar el diseño de la lógica:

El diagrama muestra una interfaz de usuario con un diseño limpio y funcional. En la parte superior, hay tres campos de entrada de texto etiquetados como 'Nombre', 'Materia' y 'Palabras clave'. Debajo de estos campos, hay un área grande de texto etiquetada como 'Texto'. En la parte inferior, hay dos botones: 'Guardar' y 'Cancelar'. Todo el contenido está encerrado en un recuadro rectangular con un borde negro.

Grafico 5: Diseño interfaz modulo *escribir*

La figura anterior es el diseño de la plantilla, donde todos los campos hacen parte del control. El botón guardar dispara la lógica del modelo, guardando en la base de datos; tabla *artículos* los datos aquí entrados. Desde luego la plantilla esta totalmente ausente de botones o algún tipo de campo, simplemente son tablas, colores, medidas y nombres de variables donde posteriormente serán remplazadas por los componentes de control. Esto último es realizado por el archivo *vista.php* cuyos formularios invocan las funciones de *control.php*, y a su vez todo proceso de manipulación de información codificado en *modelo.php* es llamado por *control.php*. En este caso concreto las validaciones son hechas por el control y el almacenamiento en la base de datos es realizado por el modelo.

## 5.2 VIABILIDAD DEL SISTEMA

**5.2.1 Viabilidad técnica.** En cuanto a este aspecto hay dos razones muy validas para sustentar la viabilidad técnica del CMS. Como primera medida es un sistema creado para cálculo y manejo de información sobre la red, lo que implica que deja de ser simplemente hipertexto representado con etiquetas, para incursionar en lenguajes de programación con sentencias. Al llevar este potencial a la red, las herramientas no son ya más exclusivas sino que pueden llegar con sus soluciones a todo aquel que tenga acceso a Internet o una intranet. Con estas facilidades las herramientas de *CMS power* están disponibles para correr sobre la web sin necesidad de ser descargadas al pc, lo que quiere decir que su funcionamiento no se reserva ni a una sola institución o entidad, ni a un grupo muy reducido que han recibido de alguna forma la aplicación, pues esta constantemente on-line.

Para hacer su desarrollo y pruebas es conveniente y practico correr la aplicación sobre un solo pc, el pc donde se esta trabajando. Como es necesario interpretar un código demás al que interpreta el navegador es necesario usar el procesador de la maquina. Es posible trabajando con una maquina simuladora (servidor local), donde el servidor (Apache en nuestro caso) interpreta los códigos al ser pedidos por el navegador. Seria esta la segunda razón, pues las soluciones que propone la aplicación presentada están disponibles para trabajar, también, sobre un solo pc.

Adicionalmente se usan herramientas de software libre que son desarrolladas, algunas por muchos programadores y teniendo acceso a las demás por la módica suma de \$ 0, en la licencia. Esta tendencia además de hacer más competitivo, mas barato y menos “cachesudo” el gremio de los programadores, a favorecido la calidad. Los componentes necesarios para el funcionamiento de *CMS power* funcionan sobre todos los sistemas operativos y con los mínimos requerimientos de maquina.

Php es un lenguaje de muy alto nivel, por lo que desarrollar en él es relativamente sencillo y rápido. Esto quiere decir que la corrección es sencilla y el tiempo de utilidad de un software PHP es más amplia gracias a lo fácil de su sintaxis.

**5.2.2 Viabilidad operativa.** Se pensaba con el presente aplicativo juntar varias tendencias que se están viendo en el momento, como lo es la investigación en línea de video, audio y texto, gracias a la utilización de bases de datos y por otro lado la posibilidad de tener una cuenta propia, donde el usuario dispone de privacidad y personalización. Se tiene entonces, funciones prácticas, muy actuales y que ya tienen penetración en la población, por eso se espera que pueda hacerse a un nicho de ésta, ya que el enfoque es masivo, el aprendizaje para su utilización no es problema y la utilidad es muy buena.



## 6 MODELAMIENTO DEL SISTEMA

### 6.1 DICCIONARIO DE DATOS

En este numeral se hará una recopilación de los datos usados en el sistema, con su correspondiente descripción, de forma que el lector casual llegado a estas líneas por azar pueda tener mejor claridad sobre la terminología utilizada.

<b>articulos</b>			
Tabla creada para almacenar los artículos escritos por el usuario, mediante el editor de "El cuarto de lavado".			
Campo	Tipo	Predeterminado	Comentarios
<u>ar_id</u>	int(11)	Auto incrementable	Llave primaria de la tabla.
ar_user	int(11)		Se almacena la cedula (nit) del usuario que escribe el artículo. Campo para relacionar con tabla <i>usuarios</i>
ar_arti	text		Contiene el texto del artículo
ar_addimg	varchar(200)		String que referencia la ruta de un archivo (imagen) que haga parte del artículo.
<b>aula</b>			
Se registran los grupos de usuarios que hacen parte de una materia. Cada profesor es dueño de una materia.			
Campo	Tipo	Predeterminado	Comentarios
<u>idaula</u>	int(11)	Auto incrementable	Llave primaria
user	int(11)	0	Es la cedula (nit) del profesor que ha creado el aula
materia	varchar(30)		Nombre de la materia
member	int(11)	0	Cedula (nit) del usuario estudiante que es miembro de esa materia.
<b>files</b>			
Se guarda un registro de todos los archivos subidos al sistema.			
Campo	Tipo	Predeterminado	Comentarios
<u>idfiles</u>	int(11)	Auto incrementable	Llave primaria
nombre	varchar(60)		Nombre del archivo tal como esta almacenado en la carpeta <i>\userfiles</i> , con su extensión.
titulo	varchar(30)		Simple nombre del archivo; sin identificador único ni extensión
ruta	varchar(50)		Ruta donde esta alojado el archivo y el respectivo nombre de éste.
exten	varchar(9)		La extensión del fichero.
tam	int(11)	0	Tamaño del archivo
namearti	varchar(40)		Nombre del artículo al cual hace parte éste. Recuerde que el modulo para subir archivos es el mismo con el que subimos texto en formato txt en lugar de escribirlo con los editores. Los artículo tiene archivos adjuntos para complementar.

materia	varchar(30)		Materia al cual hace parte el archivo
wclaves	varchar(130)		Palabras claves para facilitar la búsqueda; String de palabras separadas por espacios.
user	int(11)	0	Cedula (nit) del usuario al que pertenece el fichero
aulin	int(11)	0	Id de la tabla aula, para que los miembros de un aula sean informados del artículo en caso que sea subido su docente.
publico	Binario	0	1= publico, 0= privado
<b>msg</b>			
Toda la mensajería interna, es decir entre los usuarios, es guardada en esta tabla para su posterior presentación al destinatario.			
Campo	Tipo	Predeterminado	Comentarios
idmsg	bigint(20)	Auto incrementable	Llave primaria
msg	text		Contenido del mensaje
asunto	varchar(50)		String asunto del mensaje
de	int(11)	0	Cedula (nit) del usuario que envía
para	int(11)	0	Cedula (nit) del destinatario
fecha	date	0000-00-00	
mat	varchar(30)	0	Nombre del aula donde el usuario mandó el mensaje
<b>preguntas</b>			
Se guardan las preguntas que se utilizan a la hora del registro de un usuario, ya que éste no la escribe sino que selecciona una de las predeterminadas. Esta tabla favorece el agregar borrar o editar las preguntas.			
Campo	Tipo	Predeterminado	Comentarios
idp	int(11)	Auto incrementable	Llave primaria
preg	text		String
<b>usuarios</b>			
Registro de todos los usuarios del sistema que se hacen miembros.			
Campo	Tipo	Predeterminado	Comentarios
id	int(11)	Auto incrementable	Llave primaria
tipo	set('estudiante', 'profesor', 'admin')		Identifica si el usuario es administrador, estudiante ó docente. El campo solo permite almacenar estos tres datos.
nick	varchar(30)		Apodo alternativo al nombre del usuario
nit	int(11)	0	Cedula
pass	varchar(30)		Password del usuario
pregunta	int(11)	0	Id de la tabla pregunta, para relacionar el texto de la pregunta que el usuario escogió al registrarse.
respuesta	text		String entrado por el usuario
materia	varchar(30)		Este campo es utilizado por usuarios docentes
nombre	text		Nombre y apellido del usuario
mail	text		String
countin	int(11)	0	Este campo incrementa cada vez que la sesión es abierta.
foto	text		Ruta del archivo (imagen)

Tabla 2: diccionario de datos

## 6.2 CARPETAS

El sistema se divide en carpetas así:

Framework:

- Librerías:
  - FCKeditor
  - Smarty
  - Xajax
- Plantilla
- Plantilla\_c
- Images
- Copia; de seguridad
- Peques
- 

Plugins:

- Video
- Audio

Archivos de usuarios:

- Userfiles: Se almacenan todos, absolutamente todos los archivos que suban los usuarios. Para evitar repetición en los nombres de archivo y confusión de propietario se estableció la siguiente nomenclatura: cedula\_DíaMesAño\_hora\_nombre archivo.

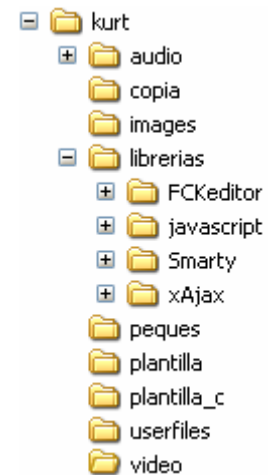


Grafico 6: Árbol de carpetas

La carpeta *librerias* contiene un sin número de herramientas –de código abierto - que no se utilizarán en *El cuarto de lavado*, sin embargo al utilizar algunas (funciones) contenidas en tales scripts es menester tener disponible todos los archivos de dicha carpeta para el correcto funcionamiento. Aquí se encuentra entre otras cosas un editor de texto escrito en JavaScript llamado FCKeditor el cual no es insertado en el aplicativo. Se aclara lo anterior con el objeto de decir al lector que el presente trabajo no profundizará en la explicación ni utilización de estas herramientas. Sin embargo se hace la siguiente aclaración en la distribución del modulo *escribir (MVC)*.

Carpetas:

- Librerías: Se encuentran las librerías donde se encuentra las funciones que hace posible la utilización de Smarty y xAjax.
- Plantilla: En esta carpeta se guarda la plantilla; *escribir.php*
- Plantilla\_c: Esta carpeta la necesita el motor *Smarty* pues allí aloja las plantillas compiladas
- Rescritor: Se aloja el modelo, la vista y el controlador. Modelo.php, vista.php, control.php

### 6.3 DIAGRAMA ENTIDAD-RELACIÓN

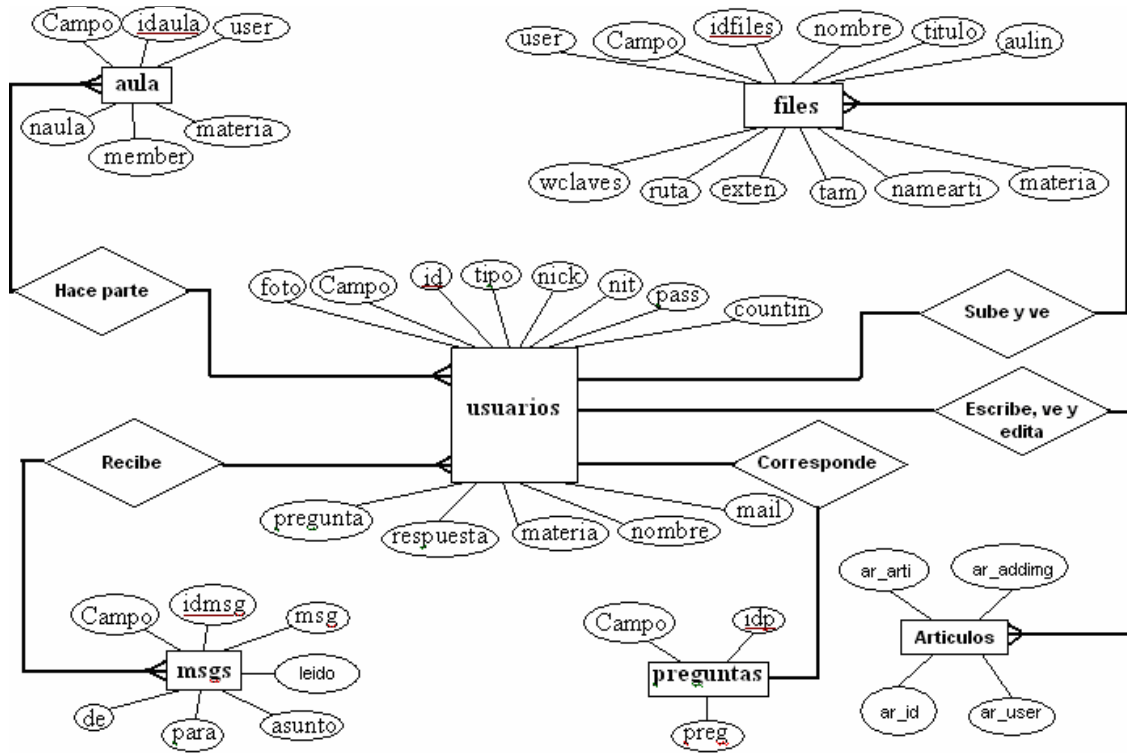


Gráfico 7: diagrama entidad-relación

## 6.4 DIAGRAMA RELACIONAL

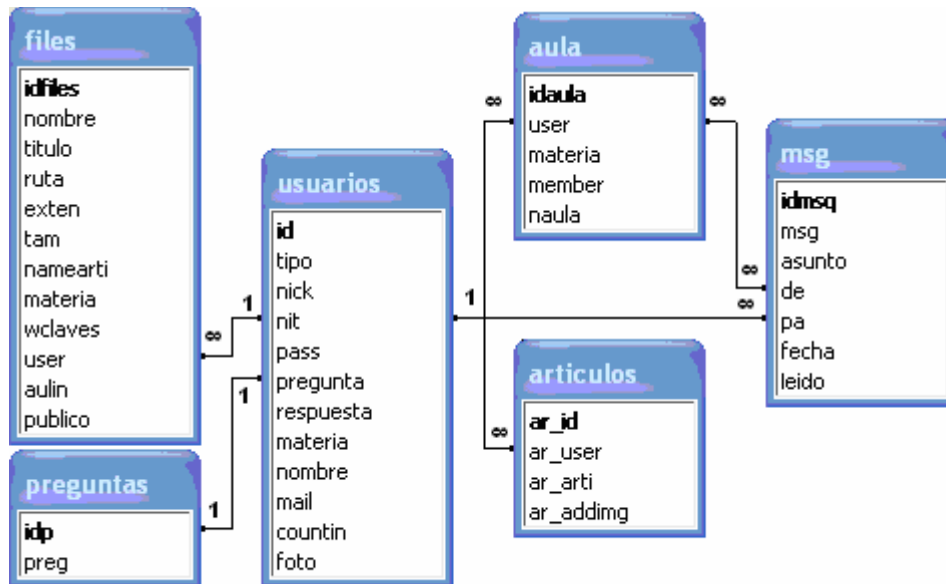


Gráfico 8: diagrama relacional

## 6.5 CASOS DE USO

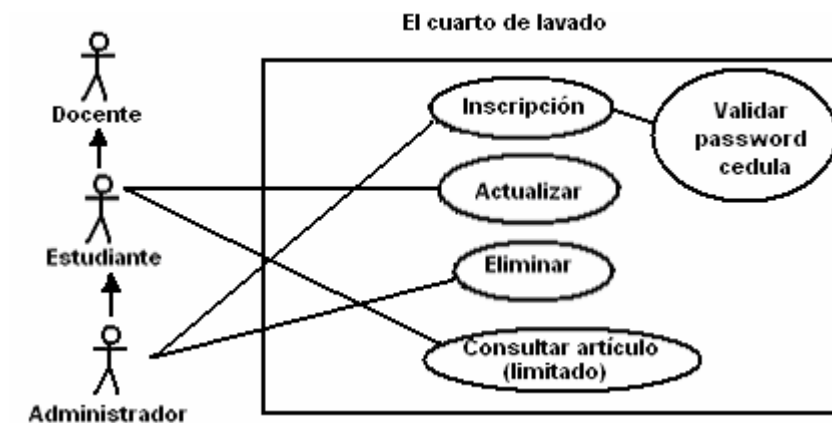


Gráfico 9: caso de uso de todos los usuarios del sistema

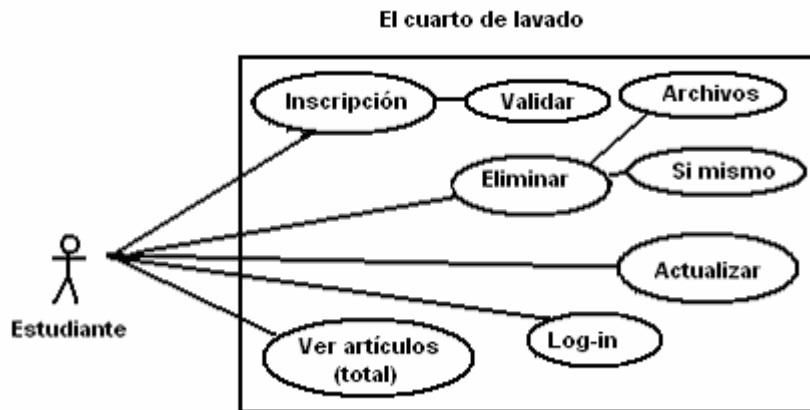


Grafico 10: Caso de uso *estudiante*

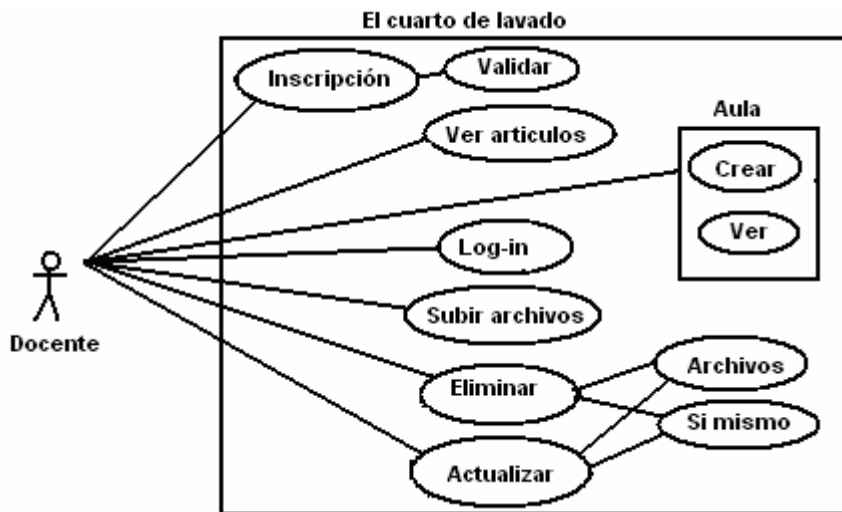


Grafico 11: Caso de uso *docente*

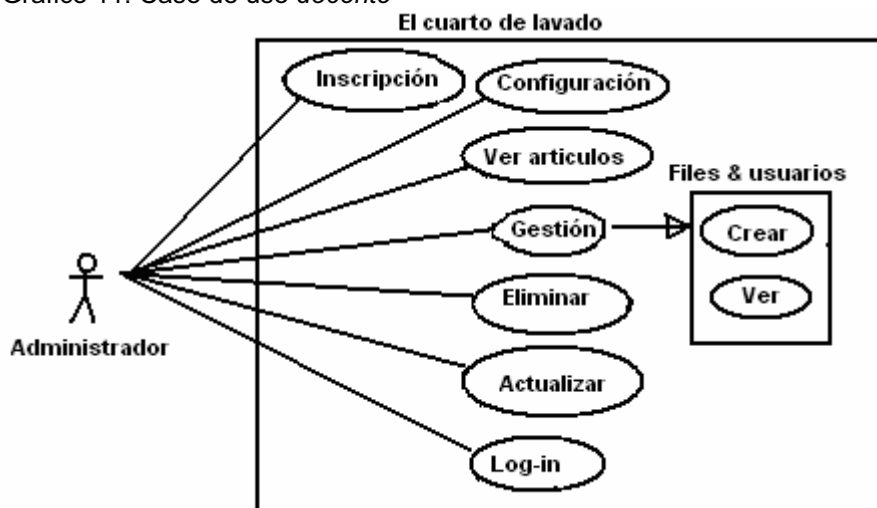


Grafico 12: Caso de uso *administrador*

## 6.6 DIAGRAMAS DE ESTADO

### 6.6.1 Diagrama de estado administrador:

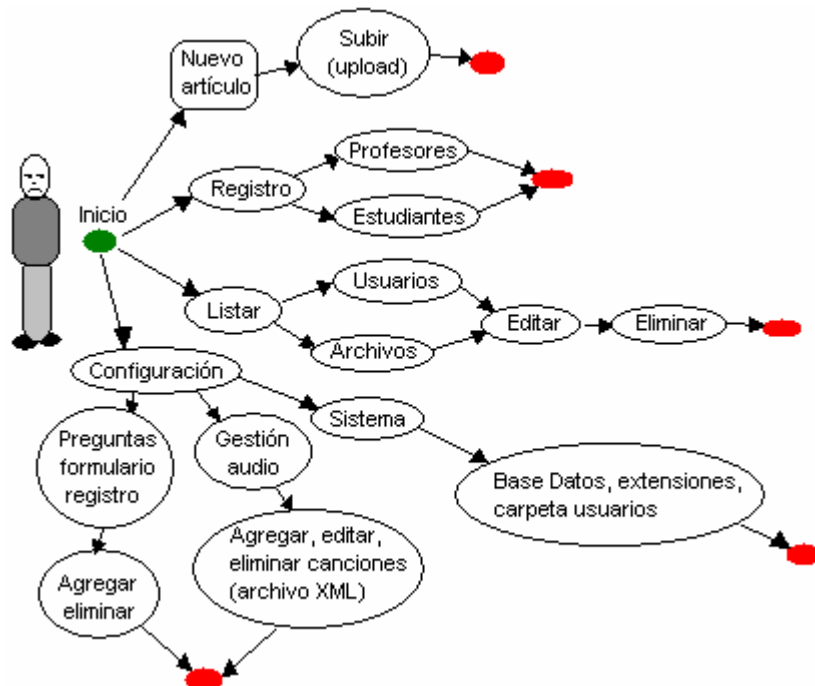


Gráfico 13: Diagrama de estado de *administrador*

### 6.6.2 Diagrama de estado Estudiante

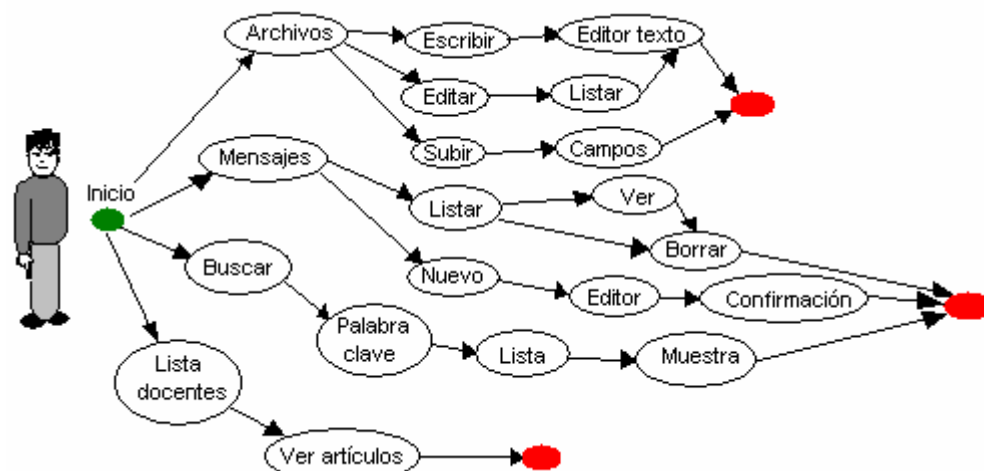


Gráfico 14: Diagrama de estado *estudiante*

### 6.6.3 Diagrama de estado profesor

El siguiente diagrama incluye también los estados del usuario estudiante, más no aparecen aquí por cuestiones de orden y espacio.



Grafico 15: Diagrama de estado *profesor*

## 6.7 DISEÑO DEL SISTEMA PROPUESTO

Ventana index: Esta ventana contiene los espacios para la entrada de usuarios registrados y botones para acceder a registrarse. Igualmente como los demás usuarios, tiene el sistema *buscador*.

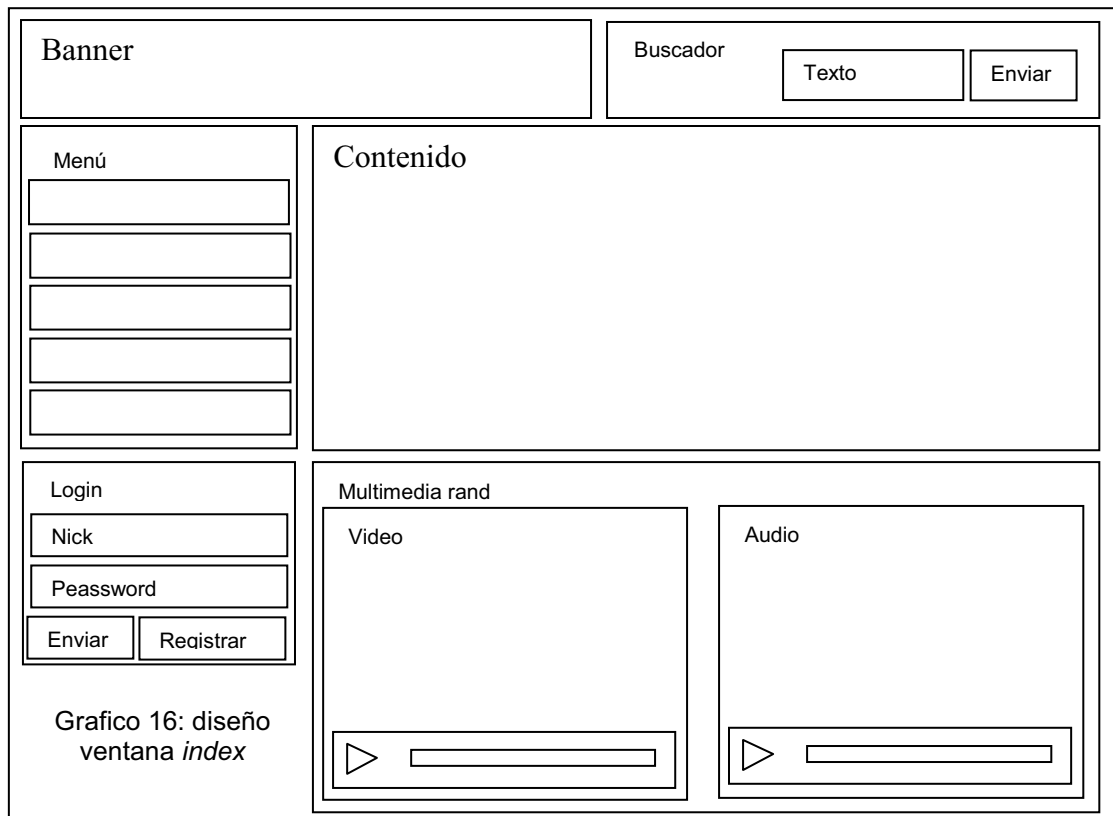


Grafico 16: diseño ventana *index*



Ventana Principal Estudiante: viene con buscador y una tala en la parte izquierda donde se le dice al estudiante los docentes que le han inscrito y los movimientos que estos hacen en cuanto a agregar artículos u archivos. En la parte inferior izquierda se encontrara las herramientas de gestión del espacio así como una imagen personalizada:

Diagrama de la ventana principal del estudiante:

- Banner**: Área superior izquierda.
- Buscador**: Área superior derecha con un campo de texto y un botón "Enviar".
- Sus docentes**: Tabla con 4 filas y 3 columnas.
- Herramientas de uso**: Área inferior izquierda que incluye un recuadro "Imagen usuario" y cuatro botones de gestión.
- Contenido y todo lo referente a presensación de consultas**: Área principal de texto a la derecha.

Grafico 17: Diseño de ventana principal *estudiante*

Ventana principal docentes: los docentes por su parte tienen herramientas para crear e inscribir estudiantes independientemente de la materia titular del docente (la que fue avisada en el momento de inscripción de este).

Diagrama de la ventana principal del docente:

- Banner**: Área superior izquierda.
- Buscador**: Área superior derecha con un campo de texto y un botón "Enviar".
- Herramientas de uso**: Área inferior izquierda que incluye un recuadro "Imagen usuario" y cuatro botones de gestión.
- Contenido y todo lo referente a presensación de consultas**: Área principal de texto a la derecha.
- Aulas**: Subsección dentro del contenido que incluye:
  - Inscribir:** Tabla con 3 columnas: Nombre, Cedula, acciones.
  - Ver aulas:** Tabla con 2 columnas: Estudiante, acciones.

Grafico18: Diseño ventana principal docente

Modulo *escribir* con modelo-vista-controlador:

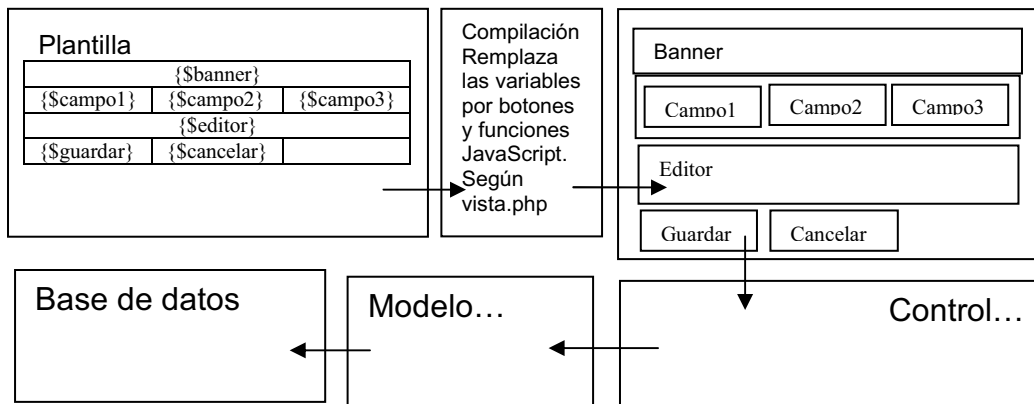


Grafico 19: diseño del módulo escribir

## 7 DESARROLLO

### 7.1 ESPECIFICACIÓN DEL SISTEMA

**7.1.1 Destinatarios de la aplicación.** Tres son los destinatarios de *El Cuarto de Lavado*. Al estar orientada a la academia, docente y estudiante juegan el papel principal. Se convierten estos en los primeros objetivos a los que va dirigido el sistema. Se tiene en cuenta también que en su gran mayoría serán usuarios con edades que oscilan entre 17 y 25 años, además de que el registro no tiene ningún costo como tampoco un acompañamiento previamente capacitado para sortear pantallazos que requieran algún tipo de conocimiento específico, por eso usuario promedio debe poder manejar intuitivamente el aplicativo, es decir saltarse la etapa de capacitación. Al ser también una herramienta de investigación, las consultas a los temas que lleguen a realizarse debe ser lo mas preciso posible (desambigüedad) pues la información tiene que llegar con los menos clicks posibles. Lo último se logra con las palabras claves, donde en primera instancia se evita investigar dentro de los artículos ahorrando tiempo y en segundo lugar, permitiendo que el usuario mismo pueda describir el tema, competencia analítica que no tiene el sistema.

### 7.2 ESPECIFICACIONES TÉCNICAS

**7.2.1 Software.** *El cuarto de lavado* es una aplicación multiplataforma por su orientación web.

Se utiliza el paquete Xampp que contiene preconfigurados MySQL, PHP, Apache, PhpMyAdmin. Es clave aclarar que los anteriores programas son los requisitos insoslayables de software para poder utilizar el aplicativo, por eso se limita a nombrar los requerimientos de sistema que necesitan dichas plataformas, ya que al final de cuentas estas mismas son las especificaciones para *El cuarto de lavado*. Estos son los requerimientos de cada uno por aparte (se darán requisitos de hardware):

- Apache 2.2.6:
  - Pentium II o equivalente
  - Linux, Free BSD, Mac OS X ó Windows NT/2000/XP
  - 9 Mb disponible.
  - 32 Mb RAM
- MySQL 5.0.45:
  - Pentium II o equivalente

- Linux, Free BSD, Mac OS X ó Windows NT/2000/XP
- 31 Mb disponibles
- 32 Mb RAM
- PHP 5.2.4
  - Pentium II
  - Linux, Free BSD, Mac OS X ó Windows NT/2000/XP
  - 15 Mb Disponibles
  - 16 Mb
- PhpMyAdmin 2.11.1:
  - Servidor Apache 1.1
  - PHP 4.7 ó superior
  - MySQL 4 ó superior
  - Internet Explorer 4, Opera 7 ó Mozilla 1.6

Se debe aclarar que para hacer correr el sistema en la maquina local, el administrador debe tener nociones básicas de Apache. Pero basta con decir que todo el conjunto de scripts, fotografías, carpetas, librerías y etcétera que conforma *el cuarto de lavado* deben estar dentro de la carpeta “www” o “htdocs” de Apache.

**7.2.2 Hardware.** Por el tipo de plataformas que se utilizaron en la construcción de *El cuarto de lavado*, no es necesario tener algún tipo de hardware especializado, pues con los recursos que actualmente poseen la mayoría de computadoras es más que suficiente. Sin embargo para información de los usuarios en el caso de que quieran saber, la aplicación puede correr desde computadoras con procesadores Pentium II o equivalentes, se recomienda Pentium III o equivalentes, 64Mb RAM y 20 Mb disponibles en disco duro.

### 7.3 ESTRUCTURA DEL CÓDIGO FUENTE

El aplicativo en su código fuente esta estructurado de diferentes formas. Se ha buscado utilizar el patrón de MVC (Modelo-Vista-Controlador) pues éste proporciona ciertas ventajas, entre ellas, facilidad en el momento de mantener ó actualizar, y un menor aclopmamiento entre módulos lo que se traduce en mayor estabilidad del sistema. Sin embargo existen lugares donde se combina la vista con el control y cuyo modelo esta en funciones del mismo script.

La forma es parecida a:

#### Login.php

```
<?
//la función que se encargará de hacer la consulta a la base de datos para la validación del login
function validaPass($nick, $pass){
    if($consul=mysql_query("SELECT `nit`,`pass`,`nick` FROM `usuarios` WHERE `pass`='$pass' AND `nick`='$nick'")){
        $tabla=mysql_fetch_array($consul);
        $_SESSION['nit']=$tabla['nit'];
        return true;
    }
    else{ return false;}
}
// La desición que disparará la función de validación, desición que solo entrará al enviar el formulario
if(isset($_POST['entrar'])){
    if(validaPass($_POST['nick'], $_POST['pass'])){
        echo "<meta content='0;URL='.$_SERVER['sever_name'].'/usuario.php' http-equiv='refresh' />";
    }
    else{
        echo "Nombre o clave incorrecta<br>";
    }
}
// Se muestra entonces el formulario al usuario
?>
<html>
<head>
<title>Log-in</title>
</head>
<body>
<form action="<? echo $_SERVER['PHP_SELF'];?>" method="post">
Nick:
<input type="text" name="nick"><br>
Clave:
<input type="password" name="pass"><br>
<input type="submit" name="entrar" value="ENTRA!">
</form>
</body>
</html>
```

Dentro de los conocimientos poseidos por los desarrolladores del *El cuarto de lavado* para la aplicación del patrón MVC se utiliza el motor de plantillas Smarty, pues permite separar fácilmente la interfaz de los procesos. También utilizamos las bibliotecas xAjax generadoras de funciones JavaScript para hacer peticiones al servidor.

La aplicación de Smarty tiene la siguiente forma:

#### Index.php

```
include('Smarty.class.php');
// Crea Objeto
$smarty = new Smarty;
// Asigna algo de contenido, este vendría comúnmente de una base de datos u otra fuente. Usaremos valores
// estáticos para este ejemplo.
$smarty->assign('nombre', 'Hernando Andres Silva Vega');
$smarty->assign('direccion', 'Cuadra podrida street');
// Seleccionamos la plantilla donde aparecerán los valores de las líneas de arriba
$smarty->display('index.tpl');
```

**Index.tlp:** Esta es la plantilla ó interfaz que verá el usuario escrita en html. En Smarty se utiliza la sintaxis "{nombre\_variable}", para que el objeto en tiempo de ejecución "pegue" los valores en dicho lugar, que en el script anterior asignamos.

```

<html>
<head>
<title>Información del Usuario</title>
</head>
<body>
<p>Información del Usuario:</p>
<p>Nombre: {$nombre}<br />
  Dirección: {$direccion}</p>
</body>
</html>

```

#### Salida HTML generada

```

<html>
<head>
<title>Información del Usuario</title>
</head>
<body>
<p>Información del Usuario</p>
<p>Hernando Andres Silva Vega<br />
  Dirección: Cuadra podrida street</p>
</body>
</html>

```

Las librerías xAjax son utilizadas para el control generalmente. Al igual que Smarty en xAjax se debe asignar, dentro del código html de la vista, el lugar donde se visualizará la información que deseamos mostrar. Al xAjax usar JavaScript utiliza un XML para hacer las operación de petición, entonces siempre se debe utilizar la función *processRequest()* para procesar las funciones y pasárselas al XML, sin embargo para hacer este proceso toda línea escrita que posea información que va a ser usada por xAjax (mostrada, enviada a otra función, etc) debe ser enviada al objeto *xajaxResponse()* donde será extraída por *processRequest()* . xAjax se usa así:

#### Config.php

```
require_once("xajax.inc.php");
```

#### Control.php

```

Include(config.php);
// Creamos el objeto xajax
$xajax = new xajax();
// toda function que se escriba con xajax debe ser registrada
$xajax->registerFunction("suma");

function suma($arg1, $arg2){
  // coge la variable $arg y haz alguna cosa con ella como realizar
  // una consulta a una base de datos y pone el resultado dentro de otra
  // variable o lo que sea
  // Haz una instancia del objeto xajaxResponse
  $objResponse = new xajaxResponse();
  $resultado=$arg1+$arg2;
  // añade un comando a la respuesta para asignar al atributo innerHTML(div)
  // del elemento con id="elemento_div" para que cambie su contenido a $resultado
  $objResponse->addAssign("elemento_div","innerHTML", $resultado);
  //retorna la respuesta XML generada por el objeto xajaxResponse
  return $objResponse->getXML();
}
$xajax->processRequest();

```

#### Vista.tpl

```

<html>
<head>
<?php $xajax->printJavascript(); ?>
</head>

```

```
<body>
<button onclick="xajax_suma(1, 2);">1+2</button>
<div id="elemento_div">Esta frase cambiará por el resultado de la función Suma </div>
</body>
</html>
```

Ahora que se ha hecho un breve resumen de cómo utilizar la estructura del código, así está distribuido:

Index.php: se encuentra dentro de la carpeta *index*

- Vista: index.php; utiliza smarty. La interfaz es inicio.html en la carpeta *plantillas*.
- Control: control.php; utilizamos xAjax
- Modelo: modelo.php; Distribuida en clases, hacemos todo lo que acostumbramos a hacer con php.

Escribir: dicho modulo esta en la carta *Rescritor*

- Vista: index.php Utiliza Smarty. La interfaz es escribir.html en la carpeta *plantillas*
- Control: control.php Usa xAjax
- Modelo: modelo.php. Distribuida en clases, allí están todas las funciones donde se desarrollan cálculos que se acostumbran a hacer con php.

estudiante1.php. Los demás módulos son al estilo estructurado; combina la interfaz, el control y el modelado de la información dentro del mismo código. Sin embargo se procura dividir las funcionalidades en diferentes scripts:

- Estudiante: estudiante1.php
- Profesor: prof1.php
- Subir archivos: articulinescrito.php; valida los archivos subidos.
- Borrado de archivos: erase.php; Es ejecutado por el estudiante, el profesor o el administrador enviando la id del registro en la base de datos. Borrar el archivo como también el registro.
- Recuperar password: forget.php; hace absolutamente todo el proceso sin variables llegadas de algún script.
- Cambio de fotografía de la sesión: foto.php
- Iniciación de todas las variables de sesión: ini.php; después de que el script index valida la sesión, se extrae de la base de datos: password, tipo de usuario, nick, nit, nombre y foto para ser almacenadas en variables de sesión, de forma que se reduzca el número de consultas en la base de datos.
- Borrar registros de la tabla aula: quitar.php; el usuario profesor puede desplegar un formulario donde hay acceso para borrar estudiantes de las aulas. Este script cumple ese objetivo.
- Formularios de inscripción: regis.php; en este script contiene los formularios en html de registro "estudiante" o "profesor", como también el código de validación de dichos campos y el código de inserción del usuario en la base de datos "usuarios".

- Buscardor: sbuscar.php; contiene el código que explora la base de datos "files" en busca de registros que puedan dar opciones al usuario que ha entrado la palabra clave.

La estructura de programación modelo-vista-controlador ha demostrado ser la más eficiente pues tiene numerosas ventajas, entre ellas:

- Facilidad para actualización del código
- División del diseño y el código
- División de los eventos y el modelado
- Programación orientada a objetos
- Reutilización de código lo que disminuye el tiempo de codificación



## 8 PRUEBAS

Este punto en particular es bien importante dentro del desarrollo del aplicativo, pues las pruebas buscan sobre todo, aunque sea un calificativo a primera vista negativo; ser destructivas para con el software. Fuera ya del escritorio del grupo desarrollador, la inclemencia para con el software no tendrá miramientos, por lo que en caso de fallas correrá el peligro de no pasar de su primera versión por no ser acogido por una comunidad de usuarios.

Simplificando la experiencia de desarrolladores metódicos podemos decir que las pruebas se destacan por cuatro cosas:

- Es la ejecución de un conjunto de instrucciones con el fin de hallar un error.
- Un buen caso de prueba es aquel que tiene alta posibilidad de descubrir un error no antes visto.
- Una prueba exitosa es aquella que devela un error.
- Se deben hacer con la menor cantidad de tiempo y esfuerzo, pues este debe ser invertido en otros pasos como desarrollo ó análisis.

Según los libros de ingeniería de software<sup>4</sup>, hay un punto dentro de la complejidad y tamaño de un software donde se hace imposible hacer pruebas exhaustivas, es decir llevarlo todo al límite (recorrer cada camino hasta descubrir un error en él) de forma que se descubran absolutamente todos los errores posibles y el aplicativo tenga la plena confianza de que funcione sin ningún fallo. No es porque no se pueda, pero en general no se tiene todo el tiempo del mundo para probar, más aún si se trata de un desarrollo robusto. Esto aumentan los costos, y el ritmo de la competencia dice que después de determinado lapso de semanas esto es un lujo.

Sin embargo, de acuerdo con el libro *Ingeniería de software; un enfoque práctico* del señor Roger Pressman tenemos dos métodos de prueba más o menos confiables que vamos a aplicar a *El cuarto de lavado*, estos son:

- Caja negra: Pruebas que se llevan a cabo sobre la interfaz del software. Se muestra que cada función es operativa al mismo tiempo que se buscan los errores. Se rectifica que la entrada, de acuerdo a lo diseñado, sea aceptada y el resultado el correcto, a la vez que la integridad de la información externa (archivos, datos) se mantenga.
- Caja blanca: Se comprueban los diferentes caminos lógicos del software proponiendo casos de prueba que ejecuten conjuntos específicos de condiciones o bucles. Ésta se hace teniendo en cuenta el código a diferencia de la prueba del numeral anterior. Se puede:

---

<sup>4</sup> Roger Pressman: Ingeniería de software; un enfoque práctico.

- Fabricar casos de prueba que ejecute al menos una vez todos los caminos del software en cuestión.
- Hacer que se ejecuten todas las decisiones en ambos sentidos.
- Se ejecuten todos los bucles en sus límites.
- Asegurar la validez de los datos.

## 8.1 PRUEBA DE CAJA BLANCA

Por la extensión que tiene *El cuarto de lavado* en el momento que se escribió este numeral se hace dispendioso hacer pruebas de caja blanca a absolutamente todo el aplicativo por razones de tiempo, sin embargo se han seleccionado las partes más importantes para aplicarle la correspondiente metodología: diagrama de grafos. Este ultimo se utiliza con el fin de representar gráficamente la complejidad ciclomatica, la cual quiere mostrar a los desarrolladores cuantos casos de prueba deben diseñar para pasar por todas las sentencias que conforman el software al menos una vez.

Diagrama de flujo: Módulo subir archivos

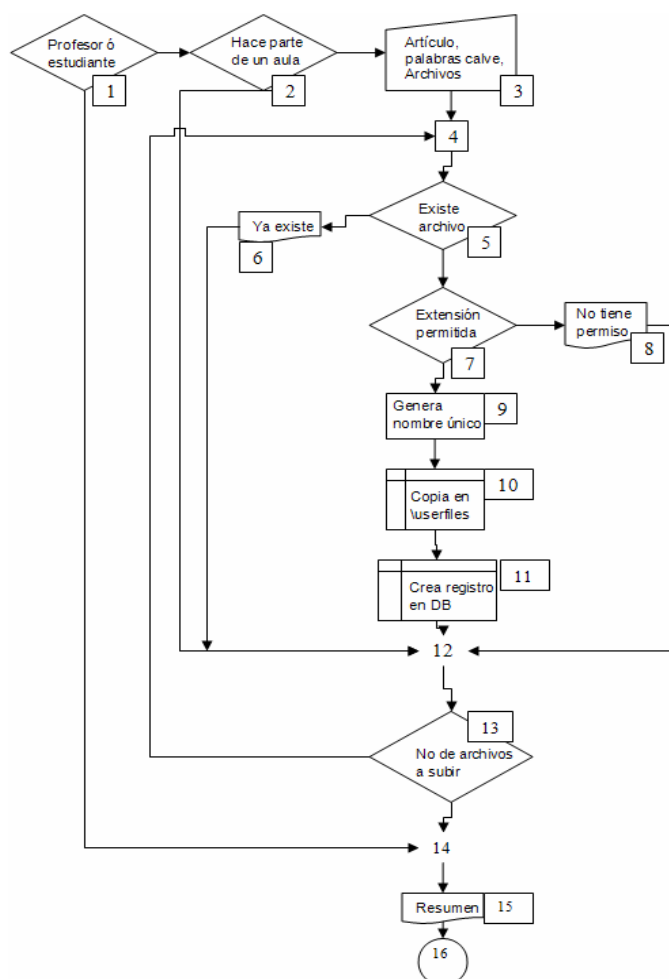


Grafico 18: diagrama de flujo *subir archivos*

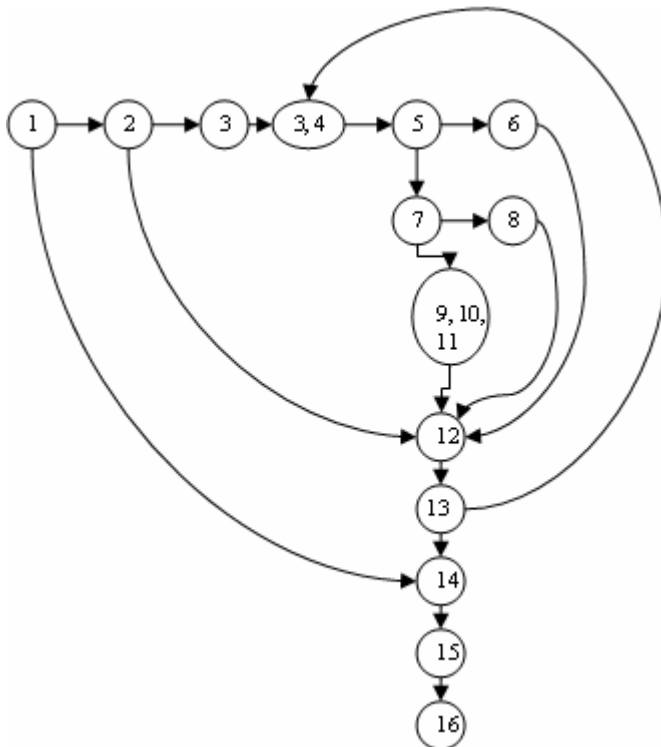


Grafico 19: Diagrama de grafos *subir archivos*

Camino 1: 1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16

Camino 2: 1, 2, 3, 4, 5, 6, 12, 13, 14, 15, 16

Camino 3: 1, 2, 3, 4, 5, 7, 6, 12, 13, 14, 15, 16

Camino 4: 1, 14, 15, 16

Camino 5: 1, 2, 12, 13, 14, 15, 16

Camino 6: 1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 3, 4, ...

Complejidad ciclomática: Éste sirve para identificar caminos independientes del conjunto básico. Da un límite mínimo de pruebas a realizar para que se ejecuten todas las sentencias al menos una vez.

$V(G) = P + 1$  donde P: Número de nodos predicados.

$V(G) = 5 + 1$

$V(G) = 6$

Se concluye que para el script *subir archivos* se necesitan al menos 6 pruebas diferentes para ejecutar todas las sentencias una vez.

Teniendo clara ya dicha metodología, haremos entonces un resumen del el resto de scripts que conforman *El cuarto de lavado*:

Sbuscar.php

Complejidad ciclomatica: 12.

Foto.php

Complejidad ciclomatica: 10

Ini.php

Complejidad ciclomatica: 4

Forget.php:

Complejidad ciclomatica: 9.

Regis.php

Complejidad ciclomatica: 10

## 8.2 PRUEBA DE CAJA NEGRA

La prueba de caja negra, tal como se ha dicho anteriormente son todo tipo de pruebas que se puedan aplicar al software desde la interfaz; esto es pruebas de validación o de comprobación de entrada y salida.

Se hicieron las pruebas sobre la interfaz. Las validaciones fue lo primero que se midió, concluyendo pues que las entradas son las correctas. En segundo lugar se miraron los datos de salida, teniendo en cuenta ya, que la información de entrada es correcta. Se hicieron las rectificaciones correspondientes. Para todo el proceso descrito abarcamos el tiempo definido dentro del cronograma, dada la extensión del proyecto, obteniendo como resultado la funcionalidad total del sistema con una correcta comunicación entre módulos y el cumplimiento de la finalidad de *El cuarto de lavado*.

## 9 GLOSARIO

**CMS:** Son las siglas en ingles de Content Management System, es decir Sistema Administrador de Contenidos. Es principalmente un gestor para páginas web donde debido a la utilización de base de datos permite cambiar en línea los diferentes tipos de contenidos que ésta contiene, además de la posibilidad de hacer cálculos y estadísticas programadas previamente sin la intervención de un individuo.

**Software libre:** Bajo el tipo de licencia GPL, se respeta la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. No es obligatorio que la distribución sea gratis, ni tampoco se debe confundir con aplicativos de dominio publico pues a diferencia de lo último éste si tiene autor que explota su producto.

**GPL:** Licencia Pública General creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

**Script:** Palabra en ingles que se usa para referirse a un código fuente escrito en cualquier lenguaje, cuya secuencia genera información que será aplicada a algo en particular. La mayoría de las veces necesita información de entrada pues a base de ella genera una salida. Un código fuente genera otro, por medio de una compilación, para que la maquina lo pueda ejecutar.

**Cachesudo:** Adjetivo utilizado en la jerga juvenil de mi generación, que señala un objeto, un comportamiento, una actitud ó un emplazamiento fino.

**Login:** Información única generada a un usuario particular para poder ser identificado por un sistema ó aplicativo, con el fin de darle un tratamiento (privilegios) diferente del resto de usuarios. Generalmente es un conjunto de información pública y privada, ésta última llamada password.

**SQL:** Structured Query Language. Es un lenguaje de cuarta generación que permite realizar consultas a bases de datos.

**MySQL:** Sistema de gestión de bases de datos relacional que utiliza lenguaje sql. Es software libre desarrollado por MySQLAB y cuenta con más de 6 millones instalaciones en todo el mundo hasta la fecha.

**Camellar:** Verbo coloquial sinónimo de trabajar duro como un esclavo, a semejanza de los camellos u otra bestia domesticada.

**Apache:** Es un programa de código abierto descargable por la red, desarrollado desde 1995. Provee servicio http incluso localmente para poder utilizar aplicaciones web.

**4GL:** Lenguaje de cuarta generación son los lenguajes en los cuales en lugar de escribir cómo deben obtenerse los resultados se especifica qué resultados son los que se quiere obtener. Por ejemplo los lenguajes de consulta de base de datos (como el SQL) son considerados lenguajes de cuarta generación.

**DB/Base de datos:** Son registros de información distribuidos en archivos. Posee campos cuyo valor, individualmente, describe una característica significativa sobre un objeto específico. Estos campos se encuentran en tablas y dichas tablas pueden relacionarse unas con otras por medio de datos comunes entre ellas, con el fin de complementar la información.

**PHP:** Lenguaje de programación orientado a la web. Usa una sintaxis y lógica semejante a c/c++. Su rapidez y capacidad se debe a que es un lenguaje del lado servidor, es decir es compilado y ejecutado por el servidor y luego el código generado presentado al usuario. La estación de trabajo no ejecuta código.

**Internet:** Red internacional que bajo el protocolo TCP/IP se comunican entre ellas. Este tipo de red no tiene un servidor central que la soporte toda, sino una numerosa cantidad de servidores y estaciones de trabajo cuyo tipo de servicio junto a una coyuntura social da su jerarquía, importancia, uso e influencia.

**LocalHost:** Nombre utilizado para la propia estación de trabajo, donde ella genera los mensajes TCP/IP y los paquetes IP para ser recibidos por ella misma. LocalHost es el nombre que señala la dirección IP 127.0.0.1, ésta no sale a Internet.

**Hardware:** Designa un componente físico de una computadora cualquiera.

**Software:** Con esta palabra nos referimos a un programa de computadora, cuyo conjunto de instrucciones permite que el usuario pueda hacer operaciones con el hardware disponible.

**XAMPP:** Paquete desarrollado por *Apache friends* que contiene, previamente configurado, las plataformas que hacen posible ejecutar los scripts generados, también, con el lenguaje PHP. Contiene: PHP 5.2, Apache 2.8, phpMyAdmin, MySQL 4, FileZilla y Mercury cuya versión, usada en el aplicativo del presente trabajo es 1.6.2.

**Navegador:** En inglés denominado browser. Es un software que permite al usuario visualizar documentos de hipertexto que llegan desde Internet, generalmente por protocolo TCP/IP. Son normalmente escritos en el lenguaje de etiquetas HTML.

**Tecnología:** Palabra de origen griego que significa arte de saberes (tekne "arte, oficio", logos "conjunto de saberes"). Este conjunto permiten fabricar objetos y modificar el medio ambiente.

**Xajax:** Es una plataforma de desarrollo escrita en PHP para generar aplicaciones en Ajax.

**Ajax:** Asynchronous JavaScript And XML (JavaScript asíncrono y XML). El usuario mantiene comunicación asíncrona con el servidor, de forma que pueda hacer cambios en la página sin necesidad de recargarla.

**Plataforma:** En computación; conjunto de librerías y aplicativos que permiten desarrollar y/o ejecutar a base o sobre ellos otros aplicativos.

**Smarty:** Motor de plantillas para PHP. Desarrollado con el fin de separar el diseño de la lógica, muy conveniente para programar bajo el esquema Modelo-Vista-Controlador. Aquí los roles del diseñador y el programador se separa en pro de mejorar la calidad del producto final, pues cada uno concentra sus energías creativas en el oficio que dominan.

**MVC:** (Modelo Vista Controlador). Esquema de programación donde se separa el código fuente en tres partes por conveniencia y agilidad. La vista es la interfaz con el usuario, el control es el conjunto de eventos que pueden ser disparados por el usuario desde la vista o por otra entidad y el modelo es el código que hace consultas y diferentes tipos de operaciones.

**XML:** Sigla en inglés de Extensible Markup Language («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML).

**SGLM:** Lenguaje que sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.

**Cookie:** es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas.

## 10 RESULTADOS OBTENIDOS

- Utilizados los varios tipos de programación hemos obtenido una aplicación donde ciertos módulos son más adaptables y actualizables que otros. Sin embargo al desglosar en diferentes archivos el código, pues esta es su estructura en su mayoría, se hace el aplicativo tan sencillo para entender, en cuanto código se refiere, casi como en un modo orientado a objetos. Por eso se hace fácilmente actualizable.
- Cada uno de los pequeños archivos que componen el aplicativo contiene código HTML para presentar al usuario el resultado de su proceso ó interfaz. Esto lo hace reutilizable para diferentes páginas o lugares dentro del sistema.
- Podemos tener en cuenta una relativa alta cohesión entre los módulos. Esto es imprescindible al pretender reducir el código dentro de los scripts con el fin de buscar estabilidad.
- El aplicativo cumple los objetivos.



## 11 CONCLUSIONES

- Se crearon interfaces que permiten manipular el sistema y lo más básico de las bases de datos, de manera que el aplicativo puede expandirse al agregarse información, archivos y usuarios.
- Se ha logrado construir e insertar con éxito herramientas de medios (multimedia) cuya utilización complementará ó sustentará mejor los artículos subidos por los usuarios, en el caso de que éstos quieran utilizarlos.
- *El cuarto de lavado* logra realizar todas las tareas que en un principio fueron determinadas, pues genera artículos y los busca; genera sesiones, maneja archivos así como también permite descargarlos o reproducirlos, sea el caso del formato. Adicionalmente tiene un usuario administrador que puede configurar las opciones más básicas del aplicativo.

## 12 SUGERENCIAS Y RECOMENDACIONES

- *El cuarto de lavado* se programó visualizando en el navegador Opera, por lo que se recomienda su utilización. Internet Explorer suele tener problemas con algunas etiquetas HTML, cosa que puede causar que *el cuarto de lavado* no se pueda usar. Dichos problemas son algunos pcs.
- En el módulo administrador se debe tener cuidado al manipular las opciones referentes a la conexión con la base de datos pues el sistema puede salir de servicio. En dicho caso se recomienda revisar la documentación.
- Para poder correr la aplicación se recomienda XAMPP 1.6.3. Este es un paquete que contiene MySQL, Apache, PHP y PhpMyAdmin, donde se instalan por medio del correspondiente gestor al ejecutar el icono de instalación. Dicha aplicación se puede conseguir en la página oficial de Apache Friends: <http://www.apachefriends.org/en/xampp.html>. Sin embargo puede funcionar con un aplicativo semejante siempre y cuando se tenga PHP 5 ó superior.

## 13 BIBLIOGRAFÍA

PRESSMAN, Roger S. Ingeniería del Software. Un enfoque práctico. 5ª Edición.

Madrid: McGRAWHIL,  
2005. 589 p.

Wikipedia enciclopedia libre (online)

<http://es.wikipedia.org/wiki/Wikipedia>

Reproductor VDS radio: Para reproducción continua

<http://www.siliconvega.com/dream/vds/>

Reproductor flv y mp3:

<http://www.videospark.com>

# 14 MANUALES

## 14.1 MANUAL DEL PROGRAMADOR

### 14.1.1 Listado de scripts.

**Admin.php:** Esta es la página del administrador. En la parte superior se encuentran las validaciones que permiten la entrada al usuario solo si se loguea desde la página index, así como la gestión con las bases de datos referentes a borrar o actualizar, no a consultar.

**conexionsql.php:** Se encuentran las sentencias que abre la conexión con mysql. Se encuentra en la carpeta *librerias*

**Configuración.php:** Biblioteca de variables, en la carpeta *librerias*; estas variables son las referentes a la conexión con las bases de datos, la ruta de la carpeta donde son alojados los archivos de los usuarios y las extensiones permitidas para los ficheros subidos.

**Erase.php:** Recibe datos de otro script para borrar archivos de la carpeta de usuarios y sus registros en la base de datos.

**Estudiante1.php:** Módulo estudiante, es todo el código que hace posible la sesión del estudiante. Éste llama a los siguientes scripts para funcionar correctamente:

- newarticulin.php
- tbl\_archivos.php
- for\_e\_c.php

**forget.php:** Este script se dispara cuando el usuario, desde el index, quiere recuperar su contraseña.

**Foto.php:** Éste modifica la imagen del usuario. Sube, borra y cambia archivos con extensiones gif o jpg, también hace lo propio con los registros de dichos archivos en la base de datos, por lo que a pesar de los que parece es relativamente complejo.

**Index.php:** Es sencillamente la página de inicio de "El cuarto de lavado". Al mismo tiempo valida el login y redirige, en caso de que la operación sea verdadera, al script ini.php que a su vez dirige al usuario a su sesión como tal. Ésta utiliza los siguientes scripts para su correcto funcionamiento:

- sbuscar.php
- ini.php
- regis.php
- forget.php
- visualArchivTxt.php
- TablaResultBusqueda.php
- videorand.php
- audiorand.php

**ini.php:** Declara variables de sesión con los valores traídos de la consulta en `index.php`, luego direcciona al usuario a su correspondiente sesión, pues ésta funciona con dichas variables.

**Musgo.php:** Esta es disparada por cualquier tipo de sesión que envíe mensajes internos. Hace la gestión respectiva a crear el registro del mensaje en la base de datos y devuelve un resultado de la operación. Como se ha dicho anteriormente, es posible mandar de una sola vez un mismo mensaje a todos los miembros de una materia.

**Newarticulin.php:** El script permite subir archivos al sistema. Hace las validaciones a los archivos, los copia en la carpeta asignada por el administrador y crea el registro en la base de datos. Luego crea un resumen del resultado de cada uno de los archivos subidos.

**Prof1.php:** Esta es la sesión del profesor. En la primera parte se encuentra la validación para permitir el acceso al usuario. Usa variables de sesión para gestionar los mensajes.

Para su correcto funcionamiento utiliza los scripts:

- `msggo.php`
- `sbuscar.php`
- `foto.php`
- `newarticulin.php`
- `quitar.php`
- `tbl_archivos.php`
- `visualArchivTxt.php`
- `visualArtiDB.php`
- `TablaResultBusqueda.php`

**Quitar.php:** Recibe datos de otros scripts para borrar estudiantes en aulas, de la base de datos.

**Regis.php:** Contiene todos los formularios para la inscripción de usuarios. Valida clave entrada al igual que todos los campos requeridos. Genera el registro en la base de datos e informa del resultado de la gestión.

**sBuscar.php:** El cometido del script es buscar la palabra clave dada en el buscador. La lógica prevé que el array a retornar no contenga ids ó campos repetidos. Trae primero consultas de la BD donde el tipo especificado en la acción de buscar (video, audio, texto, imagen, todo) coincida con el campo ``exten`` (extensión del archivo).

**Borraarticulo.php:** pop up que confirma si se desea borrar un artículo de la base de datos *articulo*, esto quiere decir que no hace ningún proceso con archivos. La lógica se desarrolla una vez el usuario pulse sobre *si*.

**For\_e\_c.php:** Para la sesión de estudiantes; son dos simples botones que disparan el módulo de escribir artículo, el cual está bajo el esquema de programación MVC. Está contenido en la carpeta *peques*.

**TablaResultBusqueda.php:** Este recibe el array de `sBuscar.php` y hace la consulta en la tabla *files*. Luego lista los resultados. Adicionalmente contiene el proceso de ocultar y mostrar las palabras claves. Está contenido en la carpeta *peques*.

**tbl\_archivos:** cuando un usuario (profesor o estudiante) pulsa sobre el botón *mis archivos*, aparece listados todos los archivos que corresponden a dicho

usuario. Para esto utiliza la variable de sesión *nit*. Este código construye la tabla.

**visualArchivTxt.php:** Este código permite mostrar el contenido de un archivo *txt* en la pagina que lo invoque.

**visualArtiDB.php:** Este muestra el contenido de un artículo puesto en la tabla *articulo*

**ChangePass.php:** Este script esta dentro de la carpeta *peques*. Es un pequeño modulo que cambia la contraseña del usuario.

**formCambioDatos.php:** Al igual que los anteriores se encuentra en la carpeta *peques*. Es un modulo que trae la información básica del usuario en un formulario para que posteriormente pueda ser actualizada.

**VideoVisual.php:** Pagina contenido en la carpeta *video*, que será disparada tan pronto se de clic sobre un archivo media (mp3 o flv), pues contiene un reproductor flash player.

**Movie.xml:** Archivo donde el reproductor lee la dirección del archivo media que debe reproducir. Este fichero es sobre escrito cada vez que se dispara esta página.

#### 14.1.2 Listado de variables de sesión

Variables de sesión para la gestión de mensajes:

`$_SESSION['msg']['carreta']:` Almacena contenido del mensaje.

`$_SESSION['msg']['asunto']:` Asunto del mensaje.

`$_SESSION['msg']['para']:` Cedula (nit) del usuario destinatario.

`$_SESSION['msg']['request']:` String que almacena el resultado de la operación.

`$_SESSION['msg']['aulin']:` Id aula desde donde fue mandado.

Dichas variables de sesión son requeridas por el sistema para la gestión de sesión se usuario y son inicializadas por ini.php. Esta variables se usan para evitar consultas y gestionar el sistema:

`$_SESSION['pass']:` Almacena el password del usuario

`$_SESSION['nick']:` Apodo del usuario

`$_SESSION['tipo']:` Estudiante/profesor/administrador

`$_SESSION['id']:` Id del usuario

`$_SESSION['nombre']:` Nombre del usuario

`$_SESSION['nit']:` Cedula del usuario.

`$_SESSION['foto']:` Ruta donde esta almacenada la foto del usuario.

## 14.2 Manual del usuario

### 14.2.1 Ingreso al sistema.

El nombre de la carpeta del aplicativo es "kurt". Allí se aloja absolutamente todo el contenido. Para el presente manual utilizaremos el paquete XAMPP por lo que las instrucciones referidas estarán orientadas a este paquete.

En el momento de tener instalado en nuestro pc la herramienta de servicio local XAMPP, debemos guardar la carpeta *kurt* en "htdocs" contenida en la carpeta raíz "xampp". Una vez allí, corriendo ya la maquina local, desde el navegador escribiremos la dirección <http://localhost/kurt> y woalaa aparecerá el primer pantallazo de "El cuarto de lavado".

The screenshot displays the user interface of the 'El cuarto de lavado' CMS. The interface includes a banner at the top with the site title and logo, a search bar on the right, and a navigation menu on the left. The main content area features a welcome message, a multimedia section with a video player, and an audio player. Callouts provide specific information about various elements:

- Menu con opciones varias:** Points to the left-hand navigation menu.
- Campo donde se despliega la información:** Points to the main content area containing the welcome text.
- Buscador de artículos en cualquier formato:** Points to the search bar on the right.
- Plug-in de video randomico:** Points to the video player showing a live performance.
- Plug-in de audio randomico:** Points to the audio player showing track information.
- Log-in para iniciar sesión:** Points to the login form with fields for 'Nick' and 'Password'.
- Link para acceder a una interfaz que recupera claves:** Points to the 'Joder! ¡¡olvide mi password!!' link.

At the bottom of the page, there is a copyright notice: "2007©".

## 14.2.2 Usuario administrador.

Por defecto el usuario administrador puede entrar; nick: admin y password: admin. Una vez adentro la siguiente es la página principal:

Banner



Buscador

  ... 

Administrador

*Registrar estudiante y profesor despliega un mismo formulario con la única diferencia del campo materia, el cual hace parte de los atributos de éste, pues para los estudiantes no es necesario dado que pueden estar inscritos en varias materias. Por otro lado la pregunta y la respuesta son campos importantes, con estos datos El cuarto de lavado puede recuperar la información de nick y password de un usuario en caso de que la haya olvidado.*

**Profesores**

Nombre:

Materia:

Nick:

Cedula:

Mail:

Password:

Confirmar password:

Pregunta:

Respuesta:

Ningún campo debe dejarse en blanco pues todos son requeridos, sin embargo en tal caso, el script correspondiente valida una vez pulsado el botón enviar y no permite la inscripción hasta que no falte ningún espacio por llenar.

Así mismo password y confirmar password son validados a la manera; que los campos deben estar llenos y coincidir entre ellos. Ésta última muestra al usuario un aviso especificando el error.

**Estudiante**

Nombre:

Nick:

Cedula:

Mail:

Password:

Confirmar password:

Pregunta:

Respuesta:

El campo cedula es obligatoria al igual que *nick*.

Dado el caso de que se ingrese un dato idéntico a uno que ya exista, no se permitirá la inscripción y se mostrará al usuario el mensaje aclarando el error.

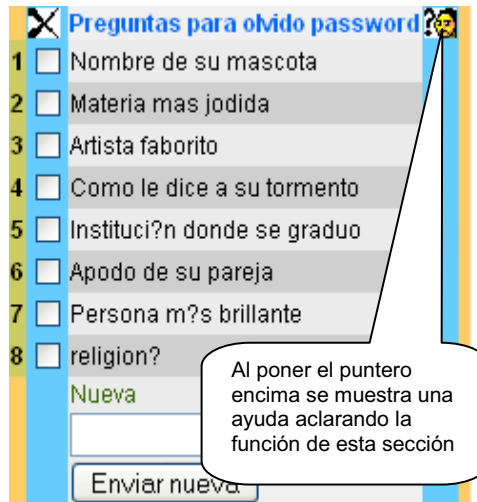


## Configuración

Al pulsar este botón se despliega las opciones  
->



Preguntas pass: Estas son las preguntas traídas de la base de datos, de acuerdo como las hayan entrado. En esta sección se pueden agregar ó eliminar según sea el capricho del administrador. Dicha sección funciona recién instalado el sistema, pues al no contar con usuario alguno, no hay oportunidad de que las preguntas seleccionadas por un usuario al inscribirse dejen de existir haciendo imposible recuperar su clave dado el caso.



Gestión audio: Aquí se muestra un campo de texto con el archivo "music.xml" para que se pueda editar. El plug-in de audio utiliza dicho archivo xml donde se le da la dirección y la descripción de las canciones a reproducir para que éste las muestre en etiquetas. Se puede agregar cuantas canciones se quiera y serán reproducidas. La sintaxis de la etiqueta es la siguiente:

```
<TRACK _artist ="nombre del artista"  
_track ="numero del track"
```

```
music_url ="ejemplo: http://www.musicapirata.com/rock/80s/WhiteLion-BrokenHeart.mp3"
```

```
link ="Url de un hipervínculo, ejemplo: http://www.google.com "
```

```
link_text ="Label del hipervínculo de arriba, ejemplo: entra a google"/>
```

Es preciso aclarar que cualquier tipo de error en la sintaxis o en la dirección del archivo provocará que el reproductor no funcione ó no se reproduzca la canción. No debe faltar ni una comilla y debe ser exacto a como aparece.

Lista canciones: [music.xml](#)

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!-- DreamedArtsStandard2005 -->  
<!-- AYUDA: Como ves, dentro de cada nodo TRACK corresponde a una pista  
de audio en tu VDS radio.  
Copia y pega el modelo de debajo tantas veces como temas vayas a añadir:  
  
<TRACK _artist =""  
_track =""  
music_url =""  
link =""  
link_text =""/>  
  
-->  
<InfoHT>  
  <TRACK _artist ="Octavio"  
    _track ="Con verraquera"  
    music_url ="userfiles/octavio mesa - con verraquera.mp3"  
    link =""  
    link_text ="Octavio"/>  
  <TRACK _artist ="Octavio Mesa"  
    _track ="El dentista"
```

Enviar

El nombre del archivo XML aparece aquí pues se valida su existencia. En caso de no ser encontrado dirá error. Probablemente el reproductor haya dejado de funcionar.

A saber, el archivo XML esta en la carpeta raíz “kurt” para su correcto funcionamiento. En caso de que el archivo no sea encontrado existe una copia de seguridad de todo el sistema de audio en la carpeta *audio*, allí esta la documentación correspondiente de la compañía desarrolladora del plug-in.

**Sistema:** Esta ventana contiene la configuración básica de conexiones con la base de datos y las extensiones permitidas. Mucho cuidado al manipular las variables, pues puede salir del aire la base de datos y *El cuarto de lavado* quedar inutilizado.

Variable	Definir
Ruta absoluta:	<input type="text" value="http://localhost/kurt/"/>
host:	<input type="text" value="localhost"/>
usuario:	<input type="text" value="root"/>
password:	<input type="text"/>
base de datos:	<input type="text" value="lavado"/>
extensiones permitidas:	<input type="text" value="txt,doc,gif,png,mp3,mp4,"/>
Ruta archivos:	<input type="text" value="userfiles/"/>
<input type="button" value="Hecho!"/>	

Separados por comas, se entran las extensiones que el administrador permitirá a los usuarios subir. Así se restringirá el tipo de archivos manipulables por los usuarios de “*El cuarto de lavado*”.

El espacio *ruta archivos* se especifica la carpeta donde serán alojados todos los archivos de los usuarios. Por defecto el nombre es “userfiles”.

## Todos los archivos

Al pulsar sobre la opción se despliega una tabla con los campos

Todos los archivos del sistema							
Id	Nombre	Exten	Ruta	Materia	Nombre articulo	Usuario	??
53	333_11102007_15_Carta pasantias.doc	doc	<a href="#">userfiles/333_11102007_15_Carta pasantias.doc</a>	Mamadas	Recompensa	333	✘
54	333_11102007_15_eu tanatus.bt	bt	<a href="#">userfiles/333_11102007_15_eu tanatus.bt</a>	Mamadas	Post	333	✘
44	333_09102007_12_beca.bt	bt	<a href="#">userfiles/333_09102007_12_beca.bt</a>	ciencias	La excrecion	333	✘
26	666_01102007_Imagen 003.jpg	jpg	<a href="#">userfiles/666_01102007_Imagen 003.jpg</a>			666	✘
46	333_09102007_12_normal_5.jpg	jpg	<a href="#">userfiles/333_09102007_12_normal_5.jpg</a>	ciencias	La excreci?n	333	✘

Última vista del archivo, pues una vez mostrada deja de existir el registro

La ruta tiene link para acceder a los archivos, se pueden eliminar con la opción “X”. Los datos mostrados son traídos de la base de datos, sin embargo el script antes de generar la lista comprueba con el campo *nombre* que los archivos existan. En caso de no existir, en primera medida, la fila se muestra de color naranja para dar un aviso y luego se borra el registro de la base de datos, así la siguiente vez que se despliegue la tabla no volverá a aparecer.

## Nuevo artículo

Se explica en el manual de estudiante.











## Usuarios

Al pulsar sobre el botón mostrara una lista de todos los usuarios del sistema. Una lista para los profesores y otra para los estudiantes. La carpetita que aparece al lado derecho muestra los archivos que ha subido dicho usuario.











Administrador

Registrar profesor Registrar estudiante Configuración Todos los archivos Nuevo articulo Usuarios Salir

**Todos los Profesores en el sistema**









Id	In's	Nombre	Nick	Nit	Materia	Correo	Acción
2	6	cabezal	elprofe	8080	matematicas		 
6	110	Juaquin	lambe	333	fisica	elandrew2000@hotmail.com	 
12	1	Gonzalo	gonzo	8000	Programacion c++	lalala@hotmail.com	 
13	0	Perrinchi	gonxo	8002	Programacion java	lalala@hotmail.com	 
14	0	Perrinchi	panic	8003	Programacion java	lalala@hotmail.com	 

**Todos los Estudiantes en el sistema**

Id	In's	Nombre	Nick	Nit	Correo	Acción
9	0	Camilo Armando	camillin	2222	elandrew2000@hotmail.com	 
7	0	Henry ford	henrysin	555	elandrew2000@hotmail.com	 
10	6	el foqui	mierd	9999	elandrew2000@hotmail.com.co	 
11	88	Juanita	juanita	8013265	juanita@hotmail.com	 
15	0	Junior	junio	654	ffrio	 

Haga clic aquí para ver los archivos del correspondiente usuario.

**Todos los Profesores en el sistema**

Id	In's	Nombre	Nick	Nit	Materia	Correo	Acción
2	6	cabezal	elprofe	8080	matematicas		 
6	110	Juaquin	lambe	333	fisica	elandrew2000@hotmail.com	 
12	1	Gonzalo	gonzo	8000	Programacion c++	lalala@hotmail.com	 
Eliminar usuario <b>Perrinchi</b> . Password administrador: <input type="text"/>							Enviar
14	0	Perrinchi	panic	8003	Programacion java	lalala@hotmail.com	 

Para que el administrador pueda eliminar un usuario, debe hacer clic sobre la "X" y aparecerá un espacio donde debe ingresar la contraseña de administrador para que el proceso tenga éxito. Esto se hace por dos fines; 1) Dado el caso que la sesión de administrador no este siendo manipulada por el individuo correcto y 2) para evitar que sea borrado un usuario por accidente.

## Salir

El botón *salir* destruye la sesión. En php una sesión se maneja por medio de *cookies*, es decir archivos que se almacenan en la carpeta "tmp", que es usada por php para alojar ficheros temporales. Al ser ejecutada la sentencia `session_destroy()` destruye todas las variables de sesión declaradas e inmediatamente se dispara la pagina de portada.

## 14.2.3 Usuario profesor.

### Inicio

Como en todas las demás interfaces, el buscador siempre esta presente y cuenta con la misma funcionalidad. La siguiente es lo que encontrará es usuario al iniciar su sesión.

**Banner**

**El cuarto de lavado**  
CMS power

**Buscador**

Buscar

Hola

Menú principal de opciones.

Acciones:

- Aula
- Articulos
- Escribir
- Subir
- Mis archivos
- Mensajes
- Salir

Hola: Juaquin  
NIT: 333

**Editar articulo**

**Articulo**

Recompensa

En este espacio aparecerá la fotografía que el usuario decida poner.

Este espacio aparecerá todas las opciones disparadas por las opciones.

### Aula

Esta opción marca la diferencia con el usuario estudiante. Se repliega las siguientes opciones:

**Aula**

[Crear aula](#) [Ver aulas](#) [Ver contenido de aulas](#)

El primer paso debe ser *crear aulas* pues una vez creada, dentro de ella se agregaran los estudiantes. De otro modo las opciones para agregar los estudiantes (que veremos a continuación) no aparecerán.

Materia	Nombre docente:	Acción
Quimica	Juaquin	Crear

Al pulsar sobre *Ver aulas* el usuario verá las aulas que ha creado:

**Aula**

[Crear aula](#) [Ver aulas](#) [Ver contenido de aulas](#)

Materia	Docente a cargo	Acción
Mamadas	Juaquin (333)	Inscribir Ver X aula
Matematicas	Juaquin (333)	Inscribir Ver X aula
Filosofia	Juaquin (333)	Inscribir Ver X aula
Quimica	Juaquin (333)	Inscribir Ver X aula

El botón inscribir desplegará un buscador de estudiantes en el caso que el profesor no sepa la cedula del estudiante que va a inscribir, solo basta con marcar el cuadro y enviar sin escribir nada en el espacio:

Inscriba y revise a sus estudiantes miembros:

Busque estudiantes

Nit:

Mostrar todos los estudiantes:

Nombre	Nit	Correo	Accion
Juanita	8013265	juanita@hotmail.com	<input type="button" value="Quitar"/>
Camilo Armando	2222	elandrew2000@hotmail.com	<input type="button" value="Quitar"/>

Nombre	Nit	Acción
Camilo Armando	2222	<input type="button" value="Agregar"/>
Henry ford	555	<input type="button" value="Agregar"/>
el foqui	9999	<input type="button" value="Agregar"/>
Juanita	8013265	<input type="button" value="Agregar"/>
Junior	654	<input type="button" value="Agregar"/>

Estudiantes inscritos a *El cuarto de lavado*

Al seleccionar un aula, en esta columna se lista los estudiantes que están inscritos al aula seleccionada. Si desea quitar algún estudiante vera el dialogo:



**Se borrará el aula con todos sus miembros. Para borrar aula ingrese su password:**

Al querer eliminar un aula, aparecerá un dialogo donde debe ingresar su contraseña. Esto es por 1) en caso de que la sesión no este siendo manipulada por el individuo correspondiente y 2) evitar que por accidente se borre un aula.

*Ver contenido de aulas* simplemente muestra un resumen de las aulas que tiene creadas con los estudiantes que hacen parte de ella. Es una ayuda informativa.

<b>Mamadas:</b>	<b>Aula No:</b>	<b>Miembros: 2</b>
◆	Juanita (8013265)	
◆	Camilo Armando (2222)	
<b>Matematicas:</b>	<b>Aula No:</b>	<b>Miembros: 2</b>
◆	Camilo Armando (2222)	
◆	Henry ford (555)	
<b>Filosofia:</b>	<b>Aula No:</b>	<b>Miembros: 0</b>
<b>Quimica:</b>	<b>Aula No:</b>	<b>Miembros: 0</b>

Editar articulo

Articulo
<a href="#">Recompensa</a>
Post
La excrecion
<a href="#">Slayer</a>
holas
La casona del sabor
maldicion
La pailera

## Artículos

El script lista solamente los registros que en la base de datos sean archivos *txt*, *doc* ó estén descritos como *artículo*, es decir que haya sido escrito por el editor de "El cuarto de lavado". Al pulsar sobre ellos se mostrará el contenido, sin embargo si el artículo es *.doc* se descargará como un archivo cualquiera.

## Escribir

**El cuarto de lavado**  
CMS power

Titulo articulo:

Materia:  Mamadas  
Matematicas  
Filosofia  
Quimica

Palabras clave:

Usuario: Juaquin

Publico:  si  no

Titulo del artículo. Campo validado

Opción de selección que muestra las aulas creadas por el profesor, pues necesaria que el artículo tenga una clasificación

Al señalar con el puntero se vera una etiqueta que explica el porque de las palabras clave.

Separadas por espacios el usuario escribirá palabras que describan su artículo.

Al marcar "si", el artículo escrito podrá ser visto por usuarios no registrados.

Campo de texto para escribir el artículo

Cuando se quiera editar el artículo escrito se vera la misma pantalla pero con los campos llenos, de forma que al *enviar* se actualizarán las bases de datos.

## Subir

**El cuarto de lavado**  
CMS power

Propiedades

Nombre:

Materia: Mamadas (333)

Palabras clave:

Cuerpo

Texto principal:

Imagen:

Imagen:

Video:

Audio:

Audio:

Audio:

Publico: si

Sube aquí un archivo

Podrá aquí subir el usuario hasta 7 archivos para un mismo artículo al mismo tiempo, por lo demás deberá subir al menos uno.

Esta sección esta diseñada para que subas tus archivos. Todo el conjunto de archivos que subas en este momento haran parte de un mismo artículo y podrán encontrarse con ayuda del buscador, pues todos los elemento estarán relacionados con las palabras claves que pongas en el espacio de arriba. Ten en cuenta que los ficheros .bt podrán ser el texto de tu artículo, es decir se vera su contenido en la página. Por lo demás el resto de archivos alimentarán tu artículo.

La fila *propiedades* tiene la misma funcionalidad que la interfaz explicada arriba. Sin embargo ésta es mucho más poderosa que la otra pues esta diseñada para que el usuario suba archivos al sistema. La fila *cuerpo* tiene siete campos para subir archivos simultáneamente. Los ficheros harán parte del mismo artículo. Téngase en cuenta que las extensiones de los archivos están restringidas por el administrador, por lo que no cualquiera será subido. Se mostrará un resumen que informa al usuario al final del proceso, cual ha sido el resultado de la validación y el proceso.

#### 14.2.4 Usuario estudiante.

**Banner**

**El cuarto de lavado**  
CMS power

**Buscador:** [Buscar] [ ] [...]

**Profesores:** Estos son sus profesores:

Profesor	Materia	Aula	Articulos
Juaquin (333)	fisica	fisica	
cabezal (8080)	matematicas	matematicas	

**Consultas:**

Subir artículo  
Mensajes  
Mis documentos  
Salir

Nombre: Juanita

Buscador

Página estándar de una sesión Estudiante.

En este espacio se despliega la información disparada por las diferentes opciones

Este cuadro lista las materias y su docente, al cual hace parte el estudiante. Recordemos que la inscripción de un estudiante a una materia la hace el docente.

#### Fotografía

En *El cuarto de lavado* la foto del usuario se gestiona de la misma forma para estudiante que para docentes, se hace así:

Subir artículo  
Mensajes  
Mis documentos  
Salir

Nombre: Juanita

Esta sección no es una biblioteca de imágenes. Subirá al servidor solo la imagen a usar.

Actual:

Mi\_foto.jpg [Examinar...]  
[Enviar] [Quitar foto]

Al *enviar* el archivo el sistema muestra el resultado de la operación; éxito o error. En cualquiera de los casos el usuario es automáticamente dirigido a su página de sesión. Es caso que se quiera *quitar foto* aparecerá en el espacio una imagen predeterminada.



## Subir artículo

Este módulo es el mismo que utiliza el docente. Revise el manual docente.

## Mensajes

Al pulsar el botón, inmediatamente aparecerá el tablón de mensajes recibidos. Cuando los mensajes no has sido leídos su texto en el tablón aparece en negrilla.

De	Asunto	Fecha	Materia	Acción
<b>Juaquin (333)</b>	<b>for</b>	<b>2007-11-02</b>	<b>Mamadas</b>	
Juaquin (333)	como van	2007-11-02	Mamadas	

**Nuevo Mensaje** - Abrir el editor y escribir mensaje

Mensaje en negrilla: no leído

Pulse para borrar el mensaje

Pulse para abrir el mensaje

Al pulsar en abrir se despliega una interfaz como la de la imagen, donde puede regresar o borrar el mensaje

Fecha: 2007-11-06  
 Del aula: Mamadas  
 De: Juanita  
 asunto: Trabajo Freud  
 mensaje:  
 Hola gonorrea, aqu? te mando instrucciones precisas sobre la cuesti?n sexual seg?n Freud. La palabra repreci?n es clave para entender la tesis de este se?or, y a lo bien es clave para entender todo comportamineto humano en casi todo tipo de situaci?n. Esto se trata de auto-moderamiento; cuesti?n fundamental en la convivencia del individuo en sociedad: alienaci?n; a la larga esto se hace cronico por la gran cantidad de fueras que logra alcanzar.

< Tablón | X Borrar

## Opciones

Se dispara los módulos correspondientes a cambio de contraseña y actualización de los datos personales.

Id	11
Tipo	estudiante
Nick	<input type="text" value="juanita"/>
Cedula	<input type="text" value="8013265"/>
Pregunta	Nombre de su mascota <input type="button" value="v"/>
Respuesta	<input type="text"/>
Nombre	<input type="text" value="Juanita"/>
Mail	<input type="text" value="juanita@hotmail.com"/>
Materia	<input type="text" value="e"/>
<input type="button" value="Enviar"/> <input type="button" value="Cambiar password"/>	

Password actual:

Nuevo password:

Cambio efectuado.  
 Su contraseña nueva es: **654321**  
 Recuerde cuidar su contraseña y no olvidarla.



Consultas

Materia:  Mamadas

Para:  el foquiu(8080)[profesor]

Asunto:

Especio de texto del mensaje

Es importante aclarar que el uso de los checkbox es por facilidad entre la mensajería interna y para limitar a los usuarios para que se comuniquen entre compañeros de aula.

Para el uso de *Escribir artículos* y *Editar artículos* acuda a líneas arriba pues.

#### 14.2.5 Usuario sin registrar.

Las siguientes son las posibilidades que tiene *El cuarto de lavado* para los usuarios que ingresan a la página y por diversas razones no se registran para hacerse miembros. Ellos tienen la posibilidad de investigar todo tipo de artículos y archivos del sistema, mientras que estos tengan la autorización de *públicos* en el momento de subidos.

#### Buscador

Buscador

Artículos

Artículos

Video

Imágenes

Audio

...

El buscador posee dos espacios; el primero es para escribir la palabra clave que será relacionada con información de la base de datos. La segunda es un combo para filtrar el tipo de información que será presentada. El campo “...” significa

“todos los anteriores.”. Se debe aclarar que en la presente versión el buscador diferencia entre mayúsculas y minúsculas; quiere decir esto que “Nirvana” es diferente de “nirvana”.

Al escribir la palabra y enviarla, este lista los resultados obtenidos y los presenta en la primera página.

Dado el caso que la palabra no se encuentre, éste presentará la tabla con la leyenda:

Resultado de la búsqueda				
Nombre artículo	Materia	Extención	Dueño	Palabras claves
No arrojó ningún resultado con la palabra <b>nirvana</b> .				

En caso contrario se vera algo parecido a:

Resultado de la búsqueda				
Nombre artículo	Materia	Extención	Dueño	Palabras claves
<a href="#">Negative creep</a>	musica	swf	333	nirvana musica video negative creep kurt cobain grunge
<a href="#">sappy (live)</a>	musica	swf	333	nirvana sappy live kurt cobain musica rock grunge

Resultado de la búsqueda				
Nombre artículo	Materia	Extención	Dueño	Palabras claves
<a href="#">Negative creep</a>	musica	swf	333	Ocultas
<a href="#">sappy (live)</a>	musica	swf	333	Ocultas

Notese que el campo *palabras clave* tiene una flecha, esta muestra u oculta las palabras claves con las que se facilita la búsqueda. Esto es porque puede existir una fila que tenga muchas palabras claves relacionadas haciendo extensa la tabla y puede darse que el sistema encuentre muchas posibilidades. Por defecto están ocultas.

Posteriormente solo basta darle clic para abrir o bajar.

La línea *"todo.\*\_all"* puesta en el espacio para buscar trae absolutamente todos los registros de la base de datos relacionados con el filtro, es decir que la palabra clave no se tiene en cuenta.