



**UNIVERSIDAD NACIONAL DEL ALTIPLANO**

**ESCUELA DE POSTGRADO**

**PROGRAMA DE MAESTRÍA**

**MAESTRÍA EN INFORMÁTICA**



**TESIS**

**“ANÁLISIS Y DESARROLLO DE UN SISTEMA ADMINISTRADOR DE  
CONTENIDOS CMS PARA EL INSTITUTO DE ESTUDIOS DE LAS  
CULTURAS ANDINAS IDECA - PUNO 2013”**

**PRESENTADA POR:**

**LUDWIN AROCUTIPA MARCA**

**PARA OPTAR EL GRADO ACADEMICO DE:**

**MAGISTER SCIENTIAE EN INFORMÁTICA**

**MENCIÓN EN INGENIERIA DE SOFTWARE**

**PUNO, PERÚ**

**2014**

UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO

BIBLIOTECA CENTRAL  
AREA DE TESIS

Fecha Ingreso: 12 JUN 2015

Nº 0787

**UNIVERSIDAD NACIONAL DEL ALTIPLANO**  
**ESCUELA DE POSTGRADO**  
**PROGRAMA DE MAESTRÍA**  
**MAESTRÍA EN INFORMÁTICA**



**TESIS**

**“ANÁLISIS Y DESARROLLO DE UN SISTEMA ADMINISTRADOR DE  
CONTENIDOS CMS PARA EL INSTITUTO DE ESTUDIOS DE LAS  
CULTURAS ANDINAS IDECA - PUNO 2013”**

**PRESENTADA POR:**

**LUDWIN AROCUTIPA MARCA**

**PARA OPTAR EL GRADO ACADÉMICO DE:**

**MAGISTER SCIENTIAE EN INFORMÁTICA**

**MENCIÓN EN INGENIERIA DE SOFTWARE**

**PUNO, PERÚ**  
**2014**

**UNIVERSIDAD NACIONAL DEL ALTIPLANO**  
**ESCUELA DE POSTGRADO**  
**PROGRAMA DE MAESTRIA**  
**MAESTRÍA EN INFORMÁTICA**

**TESIS**

**“ANÁLISIS Y DESARROLLO DE UN SISTEMA ADMINISTRADOR DE  
CONTENIDOS CMS PARA EL INSTITUTO DE ESTUDIOS DE LAS  
CULTURAS ANDINAS IDECA - PUNO 2013”**

PRESENTADA POR:

**LUDWIN AROCUTIPA MARCA**

**PARA OPTAR EL GRADO ACADÉMICO DE:  
MAGISTER SCIENTIAE EN INFORMÁTICA  
MENCIÓN EN INGENIERÍA SOFTWARE**

APROBADA POR EL SIGUIENTE JURADO:


PRESIDENTE

:

  
.....  
Dr. BERNABÉ CANQUI FLORES

PRIMER MIEMBRO

:

  
.....  
Dr. VLADIMIRO IBÁÑEZ QUISPE

SEGUNDO MIEMBRO

:

  
.....  
M.Sc. PEDRO LEONARDO QUISPE TICONA

ASESOR DE TESIS

:

  
.....  
M.Sc. ERNESTO NAYER TUMI FIGUEROA

Puno, 06 de febrero 2014

## **DEDICATORIA**

A mi madre Joaquina Marca Zapana y mi padre Pedro Arocutipa Arocutipa por su apoyo moral e incondicional.

A mi esposa Verónica Pilar Choque Quispe e hijo Gilmar Gerard Arocutipa Choque, por la constante ayuda y fuentes de mi superación.

**Ludwin**

## **AGRADECIMIENTOS**

A la Universidad Nacional del Altiplano y a la maestría de informática por brindarme la oportunidad de lograr mi anhelo de lograr una especialidad.

A los miembros del jurado, Dr. Bernabé Canqui Flores, Dr. Vladimiro Ibáñez Quispe, M.Sc. Pedro Leonardo Quispe Ticona, en su orientación en la realización del presente trabajo.

Al M.Sc. Ernesto Nayer Tumi Figueroa, como asesor de tesis quien ha guiado con sus sugerencias y consejos el desarrollo del presente trabajo.

A los docentes de la maestría en informática quienes nos dieron a conocer sus experiencias y conocimientos en el área de ingeniería software

## INDICE GENERAL

	Pág.
DEDICATORIA.....	i
AGRADECIMIENTO.....	ii
INDICE GENERAL.....	iii
INDICE DE CUADROS.....	v
INDICE DE FIGURAS.....	vi
INDICE DE ANEXOS.....	vii
RESUMEN.....	viii
ABSTRACT.....	ix
INTRODUCCIÓN.....	1

### CAPITULO I

#### PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento del problema.....	2
1.2. Justificación del problema.....	3
1.3. Objetivos	
1.3.1. Objetivo General.....	4
1.3.2. Objetivos Específicos.....	4
1.4. Hipótesis.....	4

### CAPITULO II

#### MARCO TEÓRICO

2.1. Antecedentes de la investigación.....	5
2.2. Bases teóricas.....	9
2.2.1. Portal Web.....	9
2.2.2. Aplicaciones Web.....	15
2.2.3. Ingeniería de software.....	19
2.2.4. Antecedentes de software.....	21
2.2.5. Tecnología Cliente-Servidor.....	22
2.2.6. Servidor Web.....	26

2.2.7. Servidor HTTP Apache.....	29
2.2.8. Manejador de Base de Datos.....	32
2.2.9. Visor OnLine.....	34
2.2.10. CMS.....	35
2.2.11. LMS .....	35
2.3. Definición de términos básicos .....	36

### **CAPITULO III METODOLOGÍA**

3.1. Población.....	39
3.2. Recopilación de la información .....	39
3.3. Metodología de la investigación.....	40

### **CAPITULO IV PRESENTACION DE RESULTADOS**

4.1. Definición de requisitos de sistema .....	64
4.2. Identificación de requisitos .....	64
4.3. Catalogación de requisitos .....	65
4.4. Estudio de alternativas de solución .....	67
4.5. Modelamiento del sistema.....	68
4.6. Interfaces del sistema.....	72
 CONCLUSIONES .....	 76
 BIBLIOGRAFIA .....	 78
 ANEXOS.....	 80



## INDICE DE CUADROS

	<b>Pág.</b>
1. Diagramas UML .....	41
2. Escala valorativa de la calidad del producto ISO-9126.....	91
3. Cuadro de decisiones ISO-9126.....	96
4. Escala valorativa de ficha de evaluación ISO-9126 .....	96

## INDICE DE FIGURAS

	<b>Pág.</b>
1. Aplicaciones Web .....	18
2. Arquitectura Cliente / Servidor .....	22
3. Arquitectura física Cliente / Servidor .....	25
4. Arquitectura Servidor Web .....	28
5. Servidor Apache .....	32
6. Manejador de Base de Datos MySQL .....	33
7. Visor OnLine .....	34
8. Ciclo de vida de RUP .....	43
9. Collage de diagramas UML .....	56
10. Requisitos funcionales y no funcionales .....	66
11. Diagramas de clases .....	68
12. Diagrama de flujo .....	69
13. Diagrama de secuencia .....	70
14. Diagrama de estados .....	71
15. Pagina principal del sistema .....	72
16. Modulo de Visores OnLine .....	74
17. Instituto de estudios de las culturas andinas .....	95
18. Panel del administrador .....	95
19. Validación de contraseñas .....	96
20. Vista del panel de control .....	96
21. Describiendo el panel de control .....	97
22. Editar entrada .....	98
23. Control de anuncios .....	99
24. Escritorio .....	99
25. Cerrar sesión .....	100

## INDICE DE ANEXOS

	<b>Pág.</b>
1. Anexo I .....	81
2. Ficha de evaluación de la calidad del producto estándar ISO-9126 ....	82
3. Anexo II .....	89
4. Ficha de evaluación realizadas a los inscriptores ISO-9126 .....	90
5. Anexo III .....	97
6. Manual del sistema.....	98

## RESUMEN

El Instituto de Estudios de las Culturas Andina privilegia la valoración de las culturas andinas desde una propuesta de investigación, formación y difusión de los aportes e incidencias de la cosmovisión y teología andinas a la vida social, económica, política, religiosa y cultural de nuestros pueblos andinos. En búsqueda de mejorar su alcance, ha apostado por el mundo del Internet y conjuntamente con este proyecto se está reformulando el modo de trabajo y alcance a través de la implementación del sitio WEB administrable y la implementación de aulas virtuales para los estudios de postgrado que ofrece así como la posibilidad de mantener contacto con profesionales en sociología y antropología interesados en el desarrollo y fomentación del estudio de las culturas andinas locales e internacionales. Se hace uso del UML o Lenguaje de Modelamiento Unificado para el modelamiento de los diversos casos que se presenten durante la implementación, así como para el desarrollo en si la metodología de Programación Extrema, así considerar el desarrollo iterativo del contenido del portal WEB.

**Palabras clave:** Sistema Administrador de Contenidos, culturas andinas, Sociología, Antropología, Modelamiento, Implementación

## ABSTRACT

The Study Institute of Andean Cultures privileges the valuation of the Andean cultures since a proposal for research, formation and diffusion of the contributions and impacts of the cosmos vision Andean theology to social life, economic, political, religious and cultural about our Andean town. Searching to improve its scope, has opted by internet world and together with this project is redesigning the work way and scope through the implementation the WEB place manageable and the virtual classroom studies offers and the possibility to maintain contact with professionals in sociology and anthropology interested in development and promote the study of local Andean Cultures and international ones. It makes use of UML or Unified Modeling Language for modeling of the several cases that arise during the implementation, as well as for development which is the Extreme Programming methodology and considers the interactive development of the WEB portal content.

**Key words:** Management System, Andean Culture, Sociology, Anthropology, Modeling, Implementation.

## INTRODUCCIÓN

El desarrollo de Sistemas de Administración de Contenidos en sitios Web conocidos como CMS (Content Manager System) facilita la publicación de artículos, noticias, eventos entre otros elementos Web que se deseen usar para visualización masiva.

Existen CMS bastante avanzados como Joomla, Wordpress, Drupal entre otros el problema técnico de ellos es que la codificación alcanza los 10MB en Wordpress siendo el más compacto y el caso más extremo es Joomla que alcanza los 26MB, la implementación alcanza un grado de complejidad que puede ser optimizado con técnicas de compresión y compactación de envío de paquetes HTTP y envío de formularios locales es decir sin tener que cambiar de archivo index puede reenviarse internamente la información y lograr generar las consultas únicamente usando JavaScript y recargando los resultados en un DIV con una ID definida como BodyHtml.

JavaScript y HTML5 permiten realizar estas actualizaciones sin tener que esperar recargar la página web.

## **CAPÍTULO I**

### **PROBLEMA DE INVESTIGACIÓN**

#### **1.1. Planteamiento del Problema**

El Instituto de Estudios de las Culturas Andina privilegia la valoración de las culturas andinas desde una propuesta de investigación, formación y difusión de los aportes e incidencias de la cosmovisión y teologías andinas a la vida social, económica, política, religiosa y cultural de nuestros pueblos andinos. Vivimos tiempos de reconocimiento y revaloración de las culturas andinas; pero también somos conscientes de sistemas de exclusión que desfavorecen el desarrollo de nuestros pueblos andinos en distintos ámbitos. Este pilar de la Institución de Estudios de las culturas Andinas nos lleva a apostar por la formación de actores sociales comprometidos con la búsqueda de respuestas a los momentos de cambio que atraviesa la sociedad.

En el Instituto de Estudios de las Culturas Andinas, actualmente no cuenta con una herramienta en internet que mejora la imagen y la comunicación del instituto de la noche a la mañana. Además el **portal**

**web** le ahorra tiempo, dinero y esfuerzo, nuestro proyecto se concentrará en reunir los servicios que la institución brinda reunir además información que desee hacer pública para fines informativo y otros.

¿De qué manera contribuye el Sistema de Administración de Contenidos para el Instituto de Estudios de las Culturas Andinas - Puno 2013?

## **1.2. Justificación.**

### **1.2.1. Justificación Social.**

Porque el trabajo de investigación permitirá, dar un mejor uso a las tecnologías, sino también de otros usuarios que estén interesados en proyectos similares.

### **1.2.2. Justificación Tecnológica.**

Porque el presente trabajo de investigación, hará uso de herramientas tecnológicas en software para el desarrollo de un Sistema de Administración de Contenidos, así como también de las herramientas de hardware para su adecuada implementación.



### **1.3. Objetivos.**

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

Desarrollar un Sistema de Administración de Contenidos para el Instituto de Estudios de las Culturas Andinas.

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

- Analizar y Diseñar los servicios de la institución y adaptarlos al contenido de la Pagina Web.
- Implementar la Base de Datos del Sistema Web.
- Implementar Visores On-Line para documentos.
- Desarrollar la Interfaz de Administrador de sistema.

### **1.4. HIPÓTESIS**

El Desarrollo de un Sistema Administrador de Contenidos CMS para el Instituto de Estudios de las Culturas Andinas, contribuirá en la extensión y comunicación de esta Institución.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1. Antecedentes de la Investigación**

La Ingeniería del Software es el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales. (Bauer, 1972).

Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar y operar (funcionar) y mantenerlos. Así como también desarrollo de software o producción de software. (Bohem, 1976).

Ingeniería del Software es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software. (Zelkowitz, 1978).

Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo operación (funcionamiento) y mantenimiento del software: es decir, la aplicación de ingeniería al software. (IEEE, 1993).

La Ingeniería de Software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de este después que se utiliza. (Sommerville, 2004).

La Ingeniería de Software es una disciplina que integra el proceso, los métodos, y las herramientas para el análisis y desarrollo de software de computadora. (Pressman, 2005).

En el trabajo de investigación titulado "Diseño de un sistema de información para la gerencia de ventas de una empresa de mantenimiento y suministro de equipos analíticos de laboratorio, ubicada en Puerto Ordáz, Estado Bolívar", Cuyo objetivo general es: Diseñar un sistema de información para la gerencia de ventas de una empresa de mantenimiento y suministro de equipos analíticos de laboratorio, ubicada en Puerto Ordáz, Estado Bolívar. Llegando a la siguiente conclusión: **Primero.** La Gerencia de ventas maneja un masivo número de información de forma manual lo cual deben ser guardadas y modificadas en ciertas ocasiones, lo que se convierte en un proceso tedioso. Por esta razón se propuso el Diseño del Sistema de Información (S.I.G.V). **Segundo.** La etapa de análisis de requerimientos es fundamental en la determinación de los requisitos del

sistema. Para lograr formular los requisitos funcionales y no funcionales que ayuda a establecer la forma de operar el sistema, es necesario tomar como agente primordial a los usuarios y analistas. (ROMANELLI M., R. 2009).

Plantea elevar la funcionalidad de servicios bibliotecarios de la Biblioteca de la Facultad de Ingeniería de la U.N.T. a través de la implementación de una biblioteca virtual de tesis para de esta manera promover la difusión y el intercambio de la producción académica de los alumnos de la Facultad de ingeniería. Esta investigación nace de la interrogante: "¿por qué crear la biblioteca virtual en la facultad de Ingeniería de la U.N.T.?", y de esta pregunta se desprenden dos premisas, primero el nuevo paradigma que significa una biblioteca virtual y en segundo lugar el significado que tienen estas a nivel global, profesional y en particular para dicho país ya sea en lo cultural, educativo y científico. Esta investigación, plantea una metodología exploratoria y empírica, dado que no se siguió ningún modelo anterior y lo que esencialmente se buscó documentar el objeto de estudio. Debido a su complejidad del tema propone desarrollarla por etapas y en este estudio solo asume la primera de ellas, la cual debe proyectar la biblioteca como una organización jurídicamente constituida; con una estructura administrativa, técnica y tecnológica; con colecciones y servicios apropiados a sumisión. (HUERTO L, URIOL H, 2010).

“Sistema de control de cobranzas para la administración del Colegio de Contadores Públicos de Puno – 2012”, tesis de pregrado, Escuela profesional de **Estadística e Informática, Universidad Nacional del Altiplano Puno, Perú**. Cuyo Objetivo general es: Desarrollar e Implementar un sistema de control de cobranzas óptimo para la administración del Colegio de Contadores Públicos de Puno – 2012. Llegando a las siguientes conclusiones: **Primero**. Se analizó los requerimientos del sistema de acuerdo a los estándares de calidad de software. **Segundo**. Se diseñó e implemento una base de datos para la gestión de toda la información necesaria de los módulos de cobranzas, pagos ordinarios, multas, cuotas extraordinarias, fraccionamientos, actualizar estado de colegiados, reportes de pagos, ficha del colegiado, reporte del colegiado, gestor de usuarios (nuevos colegiados) para el fácil acceso y la administración del Colegio de Contadores Públicos de Puno. **Tercero**. Se hicieron las pruebas y la validación de software con el estándar ISO – 9126 los usuarios, con las que se demostró que el sistema en mención cumple los requisitos con una media de 99 puntos según la escala de calificación del ISO 9126. Por lo cual cumple los siguientes requisitos: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad. **Cuarto**. Se desarrolló un sistema de control de cobranzas para la administración del colegio de contadores públicos de puno, la cual cumplió con los requisitos del sistema, se validó el sistema con el estándar ISO – 9126. (FLORES, 2013).

## **Relación con el Proyecto de Investigación**

Este trabajo de investigación guarda relación con nuestro proyecto, ya que es una propuesta de una Base de Datos.

## **2.2. Bases Teóricas**

### **2.2.1. Portal Web**

Los Portales Web, que están creándose cada vez en mayor número, son similares a las tradicionales Servicios Públicos, como los servicios de préstamos de libros no se encuentran realizados en papel sino en otro soporte, en formatos de textos digitalizados, ya sea en uno u otro programa, con claves o sin claves, y algunos imprimibles, mientras que otros no dan esa opción. La sociedad moderna se enfrenta a la necesidad de crear nuevas formas de organización de los recursos de información en formato electrónico, capaces de satisfacer a un usuario cada vez más exigente, que demanda un acceso rápido a los documentos a texto completo, sin la necesidad de realizar un esfuerzo significativo por su parte. Las bibliotecas virtuales son un esfuerzo por reunir lógicamente grupos de recursos de información afines, presentes en un espacio electrónico ilimitado. Es una biblioteca que tiene la tecnología que permite llevar millones de volúmenes enriquecidos con sonido e imagen en movimiento a miles de kilómetros de distancia ocupando

solo el espacio del computador. La BV es el catálogo más viejo de la web que lo empezó. Tim Berners Lee; el creador del HTML y la web misma. La BV está administrada por voluntarios confederados individuales quienes compilan páginas de descriptores ligados para áreas particulares a las que son expertos, aunque no son los índices más grandes de la web, las páginas de la BV son reconocidas ampliamente como las guías de mayor calidad a las secciones particulares de la web. Cada voluntario es responsable por el contenido de su propia página siempre y cuando sigan ciertas líneas de guía. (Glosario) La expresión "biblioteca virtual" se utiliza para describir colecciones de recursos web (FABA PÉREZ, 2004).

➤ **Roles de una Servicio Virtual.**

Una biblioteca virtual, así como la biblioteca común, depende de diversas personas que desempeñan un rol, el cual contribuye a su buen funcionamiento y servicio. Los roles que componen a una biblioteca son los siguientes:

• **Administrador.**

Como su nombre lo indica, Es la persona encargada de administrar la biblioteca virtual, Esta persona se encarga en general, de administrar las colecciones y el nuevo material adquirido, definir los permisos de cada

usuario dentro del sistema que maneja la biblioteca, y parte de la clasificación y la catalogación.

- **Usuario.**

Este rol lo desempeña todo usuario que esté afiliado a la biblioteca, se diferencia del administrador, ya que a este usuario el administrador le ha concedido ciertos privilegios dentro del sistema por estar afiliado. También es importante notar que este usuario tendrá a su disposición proponer algún material para agregarlo al repositorio de información para que pueda ser consultado, esto es, si el procedimiento de adquisición ha sido realizado, para verificar la validez de la información del material adquirido.

➤ **Ventajas y Desventajas.**

Las ventajas del manejo de información en bibliotecas virtuales están básicamente relacionadas con el acceso a información estructurada, con criterios y métodos de búsqueda útiles. Las bibliotecas virtuales también se relacionan con la preservación de los materiales y el acceso universal a la información confiable y ordenada. Al hacer uso de una biblioteca virtual no es necesario desplazarse físicamente, ni cumplir con horarios específicos de servicio. El préstamo de



materiales no está restringido al número de volúmenes, ni al tiempo o espacio ya que los servicios son ininterrumpidos y permanentes.

Los beneficios de implementar bibliotecas virtuales los podemos dividir en tres sectores. Inicialmente tiene, Beneficios nacionales y globales:

- Promueve y facilita la expansión de la cultura en una comunidad.
- Preserva en un medio no degradable los acervos culturales y científicos generados para el beneficio social. Se hace uso eficiente de los contenidos de los materiales a través de búsquedas sencillas y eficientes.
- Promueve el uso de estándares para el manejo de información digital, incluso a nivel mundial.

Los Beneficios Institucionales son:

- Elimina duplicidad de actividades, acervos y costos.
- Promueve nuevas áreas de investigación.
- Permite el crecimiento de acervos sin demanda de espacio físico para almacenamiento ni para servicio.
- Prolonga la permanencia de documentos dentro de una colección disponible al público.

- Control total sobre la información.
- Reducción del costo de imprenta, para aquellos que editen sus propios textos.

#### Los Beneficios al Usuario:

- Confianza en el contenido de los documentos que se consultan.
- Acceso uniforme desde cualquier punto de la red sin desplazamiento a la biblioteca.
- Equidad en el acceso a la información, toda la comunidad autorizada tiene la misma facilidad y derecho de consultarla.
- Siempre habrá disposición de los materiales existentes que sean solicitados.
- Ahorro en el tiempo de búsqueda por la centralización aparente de los acervos y por la posibilidad de búsqueda en el contenido completo de los documentos.
- Acceso a información interrelacionada, es decir, posibilidad de enlaces hipertextuales, incluso con otros recursos de la biblioteca virtual.

Pero para este conjunto de beneficios también existen una serie de problemas inherentes a las bibliotecas virtuales, que aún están por resolverse y que limitan la apertura y amplitud de los servicios.

Entre los más críticos podemos mencionar la falta de estándares para el manejo de información, así como la implementación de políticas de acceso. Las bibliotecas virtuales frecuentemente necesitan restringir el acceso a ciertas partes del material principalmente por las restricciones impuestas por los derechos de quien cede los documentos, por requerir de licencias o cualquier otro tema relacionado con los derechos de autor. El problema más común por el que los editores y los autores restringen el acceso es porque requieren de un pago por el uso de su material, aunque hay otras razones como el riesgo de que se efectúen copias no autorizadas.

Los sistemas Web son el medio en donde la violación de la propiedad intelectual es más común. Las leyes de propiedad intelectual se enfocan a proteger los derechos de los autores de tal manera que haya un beneficio para el autor como recompensa al beneficio que las obras le dan a la sociedad. La información digital y las redes cómputo han hecho que la copia y distribución de obras con la propiedad intelectual sea fácil y que sus autores dejen de obtener regalías, siendo el caso más crítico, en nuestro contexto, los libros.

Dentro del almacenamiento masivo de información en la biblioteca virtual, si las colecciones son grandes, se requiere de equipos con dimensiones que permitan el adecuado

almacenamiento. Asimismo, las líneas de comunicación entre las redes son de principal importancia para la eficiencia de los servicios.

Un buen dimensionamiento de ambos factores es clave para el éxito de la implementación de una biblioteca virtual. (Allison, 2011).

### **2.2.2. Aplicaciones Web**

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, Java, ASP.NET, etc.) en la que se confía la ejecución al navegador. (Allison, 2011).

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los web mails, wikis, web logs y las tiendas en línea que son ejemplos bien conocidos de aplicaciones web. Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo,

gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo. En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente que servía como interfaz de usuario que tenía que ser instalado por separado en cada ordenador personal de cada usuario. El cliente realizaba peticiones a otro programa -el servidor- que le daba respuesta. Una mejora en el servidor, como parte de la aplicación, requería normalmente una mejora de los clientes instalados en cada ordenador personal, añadiendo un coste de soporte técnico y disminuyendo la productividad. A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, que soportan por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web. Las interfaces web tienen ciertas limitaciones en las funcionalidades que se ofrecen al usuario. Hay funcionalidades comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar y

soltar que no están soportadas por las tecnologías web estándar. Los desarrolladores web generalmente utilizan lenguajes interpretados o script en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios). Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías en el lado del servidor, como por ejemplo PHP. Como ejemplo, AJAX, es una técnica de desarrollo web que usa una combinación de varias tecnologías. (Allison, 2011).

Una ventaja significativa es que las aplicaciones web deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación web se escribe una vez y se ejecuta igual en todas partes. Sin embargo, hay aplicaciones inconsistentes escritas con HTML, CSS, DOM y otras especificaciones para navegadores web que pueden causar problemas en el desarrollo y soporte de las aplicaciones web. Adicionalmente, la posibilidad de los usuarios de personalizar muchas de las características de la interfaz (tamaño y color de fuentes, tipos de fuentes, inhabilitar Javascript) puede interferir con la consistencia de la aplicación web. Otra aproximación es utilizar Adobe Flash Player o Java applets para desarrollar parte o

toda la interfaz de usuario. Como casi todos los navegadores incluyen soporte para estas tecnologías (usualmente por medio de plugins), las aplicaciones basadas en Flash o Java pueden ser implementadas con aproximadamente la misma facilidad. Dado que ignoran las configuraciones de los navegadores, estas tecnologías permiten más control sobre la interfaz, aunque las incompatibilidades entre implementaciones Flash o Java puedan crear nuevas complicaciones. Por las similitudes con una arquitectura cliente-servidor, con un cliente "no ligero", existen discrepancias sobre el hecho de llamar a estos sistemas aplicaciones web; un término alternativo es. Aplicación Enriquecida de Internet. (Allison, 2011).

FIGURA N° 01:

APLICACIONES WEB



### **2.2.3. Ingeniería de Software**

La ingeniería del software es una disciplina o área de la informática o ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelvan problemas de todo tipo. Hoy en día es cada vez más frecuente la consideración de la ingeniería del software como una nueva área de la ingeniería y el ingeniero del software comienza a ser una profesión implantada en el mundo laboral internacional, con derechos y deberes y responsabilidades que cumplir, junto a una ya reconocida consideración social en el mundo empresarial. (Agarwal & Kumar, 2001).

La ingeniería del software trata con áreas muy diversas de la informática y de las Ciencias de la Computación, tales como la construcción de compiladores, sistemas operativos o desarrollos de Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, el mundo del derecho, la red de redes Internet y Extranet.



Ingeniería del Software es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software. (Zelkovitz, 1978).

Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar y operar (funcionar) y mantenerlos. Así como también desarrollo de software o producción de software. (Bohem, 1976).

La Ingeniería del Software es el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales. (Bauer, 1972).

Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo operación (funcionamiento) y mantenimiento del software: es decir, la aplicación de ingeniería al software. (IEEE, 1993).

La Ingeniería de Software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de este después que se utiliza. (Sommerville, 2004).

La Ingeniería de Software es una disciplina que integra el proceso, los métodos, y las herramientas para el desarrollo de software de computadora. (Pressman, 2005).

#### **2.2.4. Antecedentes de Software.**

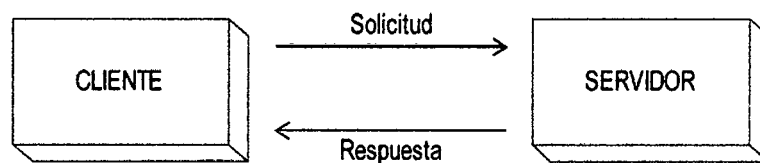
Es importante mencionar y hacer un poco de historia sobre el software que se va a utilizar ya que pertenece a la categoría del Software Libre, existe alrededor de él muchos mitos y leyendas que no son ciertos. Se mencionará la efectividad y funcionalidad que nos brinda para el desarrollo del proyecto planteado y el alcance que podemos tener de crecimiento en el ámbito de nuestras actividades con la tecnología que posee este tipo de software. No se descarta que existan herramientas comerciales con el mismo rendimiento, efectividad que ya hayan sido probados; pero también existe Software Libre de muy bajo costo, con las mismas características para resolver nuestros problemas y mejorar el funcionamiento administrativo o comercial al que estamos dedicados. Es por ello que hablare un poco sobre las características de cuatro herramientas de Software Libre que son: Linux como sistema operativo, Apache que es un servidor Web, MySQL siendo el manejador de base de datos y PHP como lenguaje de programación. Los cuáles serán usados en el desarrollo del proyecto en cuestión.

### 2.2.5. Tecnología Cliente-Servidor

El paradigma cliente / servidor es uno de los más extendidos dentro de los servicios a través de red. Este término, en su más amplia definición, se usa para describir una aplicación en la cual dos o más procesos separados trabajan juntos para completar una tarea. El proceso cliente solicita al proceso servidor la ejecución de alguna acción en particular. Esta operación se conoce como proceso cooperativo, dado que dos procesos separados cooperan para completar la tarea en particular. (Allison, 2011).

FIGURA N° 02:

ARQUITECTURA CLIENTE / SERVIDOR



## ➤ CLIENTE

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se lo conoce con el término front-end. Este normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir una interfaz gráfica de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos: (Allison, 2011).

- Administrar la Interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

### **2.2.5.1. Servidor.**

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end.

El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos: (Allison, 2011).

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de Bases de Datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de Bases de Datos.

#### **2.2.5.2. Características de un Sistema Cliente-Servidor.**

##### **Fiabilidad:**

Probabilidad de que el sistema funcione correctamente durante un intervalo de tiempo.

##### **Seguridad.**

Representa la capacidad de que el sistema no afecte ni sea afectado por su entorno y el de quien lo utiliza.

**Disponibilidad:**

Define la probabilidad de que el sistema esté funcionando en un tiempo determinado.

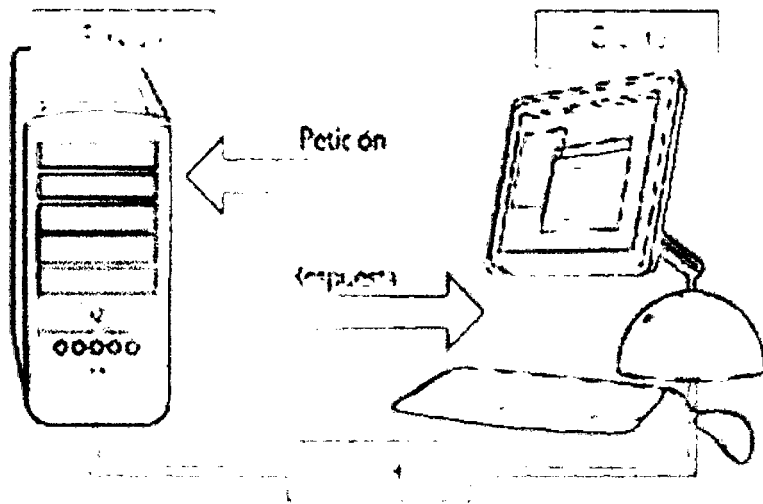
**Protección:**

Representa la capacidad del sistema para protegerse a si mismo de intrusiones accidentales o programadas.

La Disponibilidad, Seguridad y Protección se ven afectadas por la Fiabilidad.

FIGURA N° 03:

ARQUITECTURA FÍSICA CLIENTE-SERVIDOR



### **2.2.6. Servidor Web**

Un servidor web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (Hyper Text Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. Instalar un servidor web en nuestro PC nos permitirá, entre otras cosas, poder montar nuestra propia página web sin necesidad de contratar hosting, probar nuestros desarrollos vía local, acceder a los archivos de nuestro equipo desde un PC remoto (aunque para esto existen otras opciones, como utilizar un servidor FTP) o utilizar alguno de los programas basados en web tan interesantes que están viendo la luz últimamente. El problema de usar nuestro ordenador como servidor web es que conviene tenerlo encendido permanentemente (para que esté accesible de forma continua como la mayoría de los sitios webs), con el consiguiente coste debido al consumo de electricidad

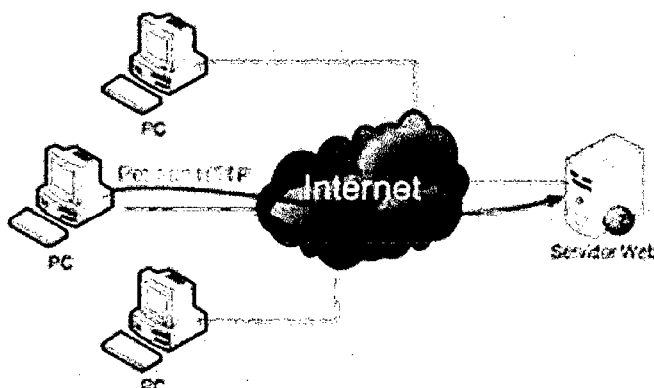
(conviene tener en cuenta que hay alojamientos web gratuitos, incluso sin publicidad y con interesantes funciones). Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un lenguaje de marcas y HTTP es un protocolo. Un servidor web se mantiene a la espera de peticiones HTTP por parte de un cliente HTTP que solemos conocer como navegador. El cliente realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear `www.wikipedia.org` en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrarlas fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma. Sobre el servicio web clásico podemos disponer de aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre: Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o JavaScript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un



navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje JavaScript y Java, aunque pueden añadirse más lenguajes mediante el uso de plugins. Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP. Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones JavaScript o Java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

FIGURA N° 04:

ARQUITECTURA SERVIDOR WEB



### **2.2.7. Servidor HTTP Apache**

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado"). El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

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. Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el

servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft). La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby). Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM Web Sphere Application Server. Mac OS X integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones Web Objetos. Es soportado de alguna manera por Borland en las herramientas de desarrollo Kylix y Delphi. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux. Apache es usado para muchas otras tareas donde el

contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos. Los programadores de aplicaciones web a veces utilizan una versión local de Apache en orden de pre visualizar y probar código mientras éste es desarrollado. Microsoft Internet Information Services (IIS) es el principal competidor de Apache, así como Sun Java System Web Server de Sun Microsystems y un anfitrión de otras aplicaciones como Zeus Web Server. Algunos de los más grandes sitios web del mundo están ejecutándose sobre Apache. La capa frontal (frontend) del motor de búsqueda Google está basada en una versión modificada de Apache, denominada Google Web Server (GWS). Muchos proyectos de Wikimedia también se ejecutan sobre servidores web Apache. La licencia de software bajo la cual el software de la fundación Apache es distribuido es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto. La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original. La Free Software Foundation no considera a la Licencia Apache como compatible con la versión 2 de la GNU General Public License (GPL), en la cual el software licenciado bajo la Apache License no puede ser integrado con software distribuido

bajo la GPL. Sin embargo, la versión 3 de la GPL incluye una provisión (Sección 7e) que le permite ser compatible con licencias que tienen cláusulas de represalia de patentes, incluyendo a la Licencia Apache. El nombre Apache es una marca registrada y puede ser sólo utilizada con el permiso expreso del dueño de la marca (Allison, 2011).

FIGURA N° 05:  
SERVIDOR APACHE.



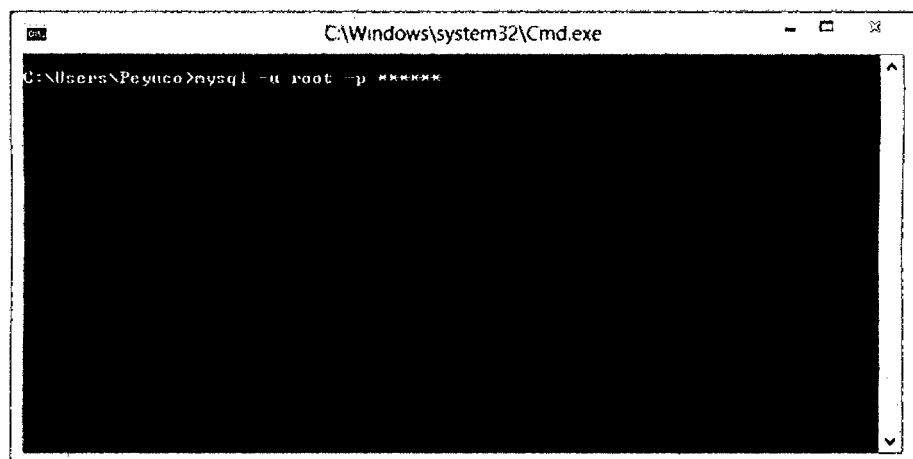
#### **2.2.8. Manejador de Base de Datos: MYSQL.**

Una base de Datos es un conjunto de datos estructurados. Esto podría ser cualquier cosa, desde una simple lista de productos, hasta una galería de imágenes, o la gran cantidad de información que se maneja en las bibliotecas de la UNAM, por dar un ejemplo. Para agregar, acceder y procesar los datos almacenados en un medio computacional, se requiere de un sistema administrador de base de datos como MySQL, conocido también como un lenguaje de consulta

estructurado, y es uno de los Motores de base de datos que acompaña a la mayoría de las distribuciones de LINUX en la actualidad. MySQL está catalogado como un software Open Source o mejor dicho de Fuente Abierta, esto quiere decir que cualquiera puede tener o contar con el código fuente y hacer modificaciones, sin la necesidad de pagar licencias de uso. Y esto es otra de las características que hacen a MySQL un Motor sumamente atractivo. Además desde su sitio oficial en Internet se puede descargar gratuitamente. MySQL es muy rápido, seguro y fácil de usar y aunque está bajo un desarrollo constante, MySQL ofrece un conjunto de funciones muy poderosas. La conectividad, velocidad y seguridad hace de MySQL una suite poderosa, para acceder a Base de Datos en Internet. (Allison, 2011).

FIGURA N° 06:

MANEJADOR DE BASA DE DATOS. MYSQL.

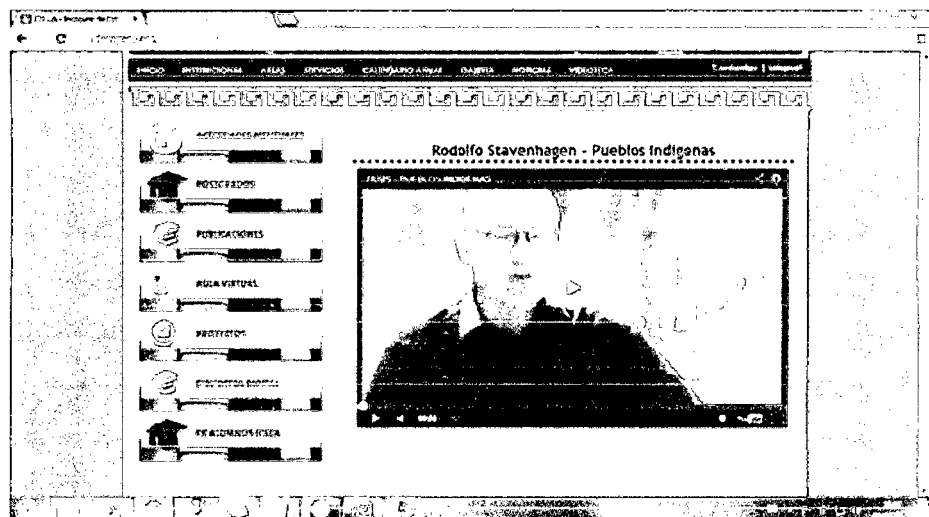


### 2.2.9. Visor OnLine

Es el intercambio de formatos de documentos no editables como PDF, PostScript, XML-View, RTF, y DOC, soportados por los distintos navegadores web, es decir no requieren instalar gran cantidad de software únicamente un plugin que se agrega al instalar el visor de documentos PDF sea cual sea la versión y empresa que la desarrolle (Allison, 2011).

En la **figura 7** podemos observar como se muestra un visor OnLine embebido en nuestra pagina Web, no requerimos salir de nuestra ventana sino en el código se agrega unos especificadores que permitan que el contenido multimedia audio, video o documentos se muestren en la misma pagina web sin alterar el funcionamiento de esta. (Allison, 2011).

FIGURA N° 07:  
VISOR ONLINE PARA VIDEOS DE YOUTUBE EN NUESTRO  
WEBSITE



### **2.2.10. CMS**

Los Sistemas de Administración de Contenidos permiten dejar la capa de desarrollo HTML y CSS bajo un nivel que no requiere conocimientos técnicos sobre el desarrollo, más que un usuario y contraseña para poder administrar Grupos, Categorías, Noticias, Artículos y otros elementos Web administrados sobre categorías personalizables y anidadles. (Allison, 2011).

### **2.2.11. LMS**

Los Sistemas de Administración de Aprendizaje (Learning Management System) son esquema CMS orientados al aprendizaje virtual uno de los primeros fue Moodle y actualmente uno de los más funcionales es Chamilo ambos desarrollados bajo la orientación de software libre para lograr el máximo de su desarrollo e incrementar la productividad del mismo. Una de las características principales de los sistemas de administración de aprendizaje es el manejo de sesiones de clases, administración de cursos, profesores, control de asistencias entre otros. (Allison, 2011).



## **2.3. Definición de Términos Básicos**

### **2.3.1. Servicio Virtual**

Los Servicios Virtuales son bases de datos y servicios en Internet para la consulta de diversos servicios de consulta, contenidos HTMLy PDF sobre plataformas web con la capacidad de poder realizar búsquedas y consultas de sobre el contenido de cada elemento.

### **2.3.2. Aplicaciones Web**

Es el servicio final conjuncionado de tecnologías HTML y el modelamiento en un software sobre un problema que se da solución como software.

### **2.3.3. Ingeniería del Software**

La ingeniería de Software es la disciplina que normaliza los procesos de desarrollo de software, incluyendo el análisis de sistema de información y el modelamiento de dichos sistemas informáticos.

### **2.3.4. Sistema Operativo: LINUX.**

Linux es un sistema operativo de libre distribución, desarrollado por la comunidad GNU bajo la licencia GPL que permite modificar el

código fuente con la condición de publicar los cambios documentarlos y detalla el proceso.

### **2.3.5. Servidor.**

Es un equipo o Hardware diseñado para brindar servicios basados en HTTP para proveer intercambio o producción de información bajo criterios de los protocolos de aplicaciones.

### **2.3.6. Servidor Web**

Un servidor web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo permite que el internet basado en este protocolo se mantenga libre en la mayoría de servidores.

### **2.3.7. Servidor HTTP Apache**

Apache es un servidor WEB gratuito, desarrollado por aficionados actualmente soportado incluso por Oracle para su distribución Tomcat.

### **2.3.8. Manejador de Base de Datos: MYSQL.**

Gestor de bases de datos libre, desarrollada en los 90s y fiable al ser rápida y soportada por la mayoría de plataformas basadas en Unix y Linux.

### **2.3.9. Lenguaje PHP.**

Lenguaje a ser interpretado para generar HTML, interpretado por los traductores PHP que funcionan del lado del servidor.

### **2.3.10. Base de Datos.**

Es una colección de Tablas, registros y campos organizados en un esquema de jerarquía que permite almacenar y ordenar información caracterizando los elementos internos de una entidad o grupo de elementos que generaran una multitud.

### **2.3.11. Sistemas Gestores de Bases de Datos (SGBD).**

Los gestores de bases de datos son los programas que administran la información, indexándola y ordenándola y administrada para que una segunda capa que es usualmente un IDE de programación haga consultas de la información contenidas y regresándola en una tabla o recordset.

## **CAPÍTULO III**

### **METODOLOGÍA**

#### **3.1. Población**

La población de quienes validaron el sistema estuvo conformada por los operadores del sistema, en un total de 3. El personal del Instituto de Estudios de la Culturas Andinas encargado del mantenimiento del portal Web.

#### **3.2. Recopilación de la Información**

Para la recopilación de datos se utilizó el siguiente instrumento que fue aplicado en los inscriptores autorizados por el personal del Instituto de Estudios de la Culturas Andinas encargado del mantenimiento del portal Web.:

- La ficha de evaluación del ISO-9126 que ve la calidad del software.

### **3.3. Metodología de Investigación.**

#### **3.3.1. RUP (Proceso Unificado de Rational).**

Es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado (UML) constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

UML está compuesto por los siguientes diagramas:

**CUADRO N° 01**  
**DIAGRAMAS UML**

<b>Area</b>	<b>Vista</b>	<b>Diagramas</b>	<b>Conceptos Principales</b>
<b>Estructural</b>	<b>Vista Estática</b>	<b>Diagrama de Clases</b>	<b>Clase, asociación, generalización, dependencia, realización, interfaz.</b>
	<b>Vista de Casos de Uso</b>	<b>Diagramas de Casos de Uso</b>	<b>Caso de Uso, Actor, asociación, extensión, generalización.</b>
	<b>Vista de Implementación</b>	<b>Diagramas de Componentes</b>	<b>Componente, interfaz, dependencia, realización.</b>
	<b>Vista de Despliegue</b>	<b>Diagramas de Despliegue</b>	<b>Nodo, componente, dependencia, localización.</b>
<b>Dinámica</b>	<b>Vista de Estados de máquina</b>	<b>Diagramas de Estados</b>	<b>Estado, evento, transición, acción.</b>
	<b>Vista de actividad</b>	<b>Diagramas de Actividad</b>	<b>Estado, actividad, transición, determinación, división, unión.</b>
	<b>Vista de interacción</b>	<b>Diagramas de Secuencia</b>	<b>Interacción, objeto, mensaje, activación.</b>
		<b>Diagramas de Colaboración</b>	<b>Colaboración, interacción, rol de colaboración, mensaje.</b>
<b>Administración o Gestión de modelo</b>	<b>Vista de Gestión de modelo</b>	<b>Diagramas de Clases</b>	<b>Paquete, subsistema, modelo.</b>
<b>Extensión de UML</b>	<b>Todas</b>	<b>Todos</b>	<b>Restricción, estereotipo, valores, etiquetados.</b>

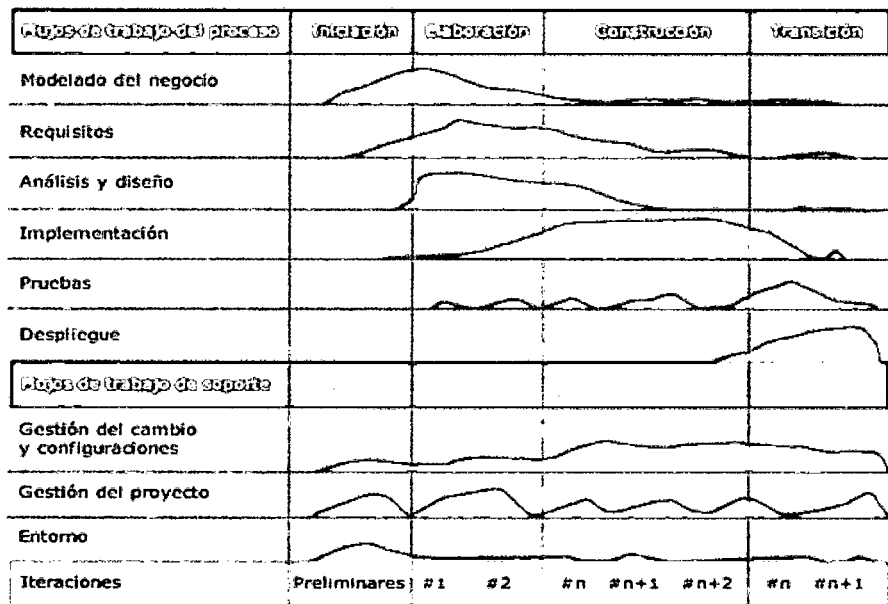
RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

**Características:**

- **Dirigido por casos de uso:** Los casos de uso se utilizan como un artefacto básico para establecer el comportamiento deseado del sistema, para verificar y validar la arquitectura del sistema, para las pruebas y para la comunicación entre las personas involucradas en el proyecto.
- **Centrado en la arquitectura:** La arquitectura del sistema se utiliza como un artefacto básico para conceptuar, construir, gestionar y hacer evolucionar el sistema en desarrollo.
- **Un proceso iterativo:** Permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema.
- **Reduce el costo de riesgo:** Si los desarrolladores tienen que repetirla iteración, la organización solo pierde el esfuerzo mal empleado de la iteración. No el valor del producto entero.
- **Un proceso incremental:** es aquel que involucra la continua integración de la arquitectura del sistema para producir esos ejecutables, donde cada nuevo ejecutable incorpora mejoras incrementales sobre otros.

- **RUP es un proceso configurable.** RUP es adaptable y puede configurarse para cubrir las necesidades del proyectos que van desde pequeños equipos de desarrollo hasta grandes empresas de desarrollo.
- **RUP impulsa un control de calidad y una gestión del riesgo.** La evaluación de la calidad va contenida en el proceso, en todas las actividades e implicando a todos los participantes, mediante medidas y criterios objetivos.

FIGURA N° 8:  
CICLO DE VIDA DE RUP



En el ciclo de vida RUP veremos una implementación del desarrollo en espiral. Con el ciclo de vida se establecen tareas en fases e iteraciones. El RUP maneja el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable. Las primeras



iteraciones (en las fases de inicio y elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una base de inicio.

### **3.3.2. Fases de la Metodología RUP**

- **Inicio**

Se establece el modelo del negocio para el sistema y se delimita el alcance del proyecto.

- **Elaboración**

Se analiza el dominio del problema, establecer un seguro funcionamiento arquitectónico, desarrollar el plan del proyecto, y eliminar los elementos de riesgo más altos del proyecto.

- **Construcción.**

Se desarrollan todos los componentes restantes y se terminan los acabados de la aplicación y se integran en el producto, probándose completamente.

- **Transición**

Se requiere típicamente que algún subconjunto utilizable del sistema se ha completado a un nivel aceptable de calidad y la documentación está disponible para el usuario para que la transición al usuario tenga resultados positivos.

### **3.3.3. Elementos de la Metodología RUP.**

#### **WORKER: EL QUIÉN**

Define la conducta y responsabilidades de un individuo, o un grupo de individuos que trabajan juntos como un equipo.

#### **ACTIVIDAD: EL CÓMO**

Es el resultado parcial o final que es producido y usado durante el proyecto. Son las entradas y salidas de las actividades.

#### **ARTEFACTO: EL QUÉ**

Es una parte de información que se produce, modifica, usado por un proceso. Los artefactos son los productos tangibles del proyecto.

#### **WORKFLOWS: EL DÓNDE**

Es una sucesión de actividades que producen un resultado de valor notable. En términos de UML, un workflow puede expresarse como un diagrama de sucesión, un diagrama de la colaboración, o un diagrama de actividad.

#### **3.3.4. Modelo del Negocio del Sistema.**

El Proceso Unificado Rational se dirige proporcionando un idioma común para ambas comunidades, así mismo muestra cómo crear y mantener trivialidad directa entre el negocio y modelos del software. Los casos del uso de negocio se analizan para entender cómo el negocio debe apoyar los procesos de negocio.

#### **3.3.5. Requerimientos del Sistema.**

La meta de los Requisitos de Workflow es describir lo que el sistema debe hacer y debe permitir a los diseñadores y clientes estar de acuerdo en las descripciones. Se identifican a los actores y se representan a los usuarios, y a cualquier otro sistema que puede actuar recíprocamente con el sistema a desarrollarse. Use-case identifican y representan la conducta del sistema.

#### **3.3.6. Análisis y Diseño del Sistema.**

La meta del workflow en el Análisis y Diseño es mostrar cómo el sistema se comprenderá en la fase de aplicación. Se debe construir un sistema que cumple todos sus requisitos.

Se estructura para ser robusto (fácil cambiar si y cuando sus requisitos funcionales cambian). El modelo del diseño consiste en clases del diseño estructuradas en los paquetes del diseño y subsistemas del

diseño con interfaces bien definidas y representa lo que se volverá componentes en la aplicación.

➤ **Implementación del Sistema.**

El propósito de aplicación es:

- Definir la organización del código, en los términos de subsistemas de aplicación organizados en capas.
- Para llevar a cabo clases y objetos en términos de componentes (los archivos fuentes, binarios, ejecutables, y otros).
- Para probar los componentes desarrollados como unidades.
- Para integrar los resultados producidos por implementaciones individuales (o equipos), en un sistema ejecutable.
- El sistema se comprende a través de la aplicación de componentes.

➤ **Prueba del Sistema.**

Los propósitos de comprobación son:

- Para verificar la interacción entre los objetos.
- Para verificar la integración apropiada de todos los componentes del software.

- Para verificar que todos los requisitos se han llevado a cabo correctamente.
- Identificar y asegurar defectos se dirigen antes del despliegue del software.

➤ **Despliegue del Sistema.**

El propósito del workflow del despliegue es producir el producto con éxito, y entregar el software a sus usuarios finales.

Cubre una amplia gama de actividades:

- Produciendo descargos externos del software.
- Empaquetando el software.
- Distribuyendo el software.
- Instalando el software.
- Proporcionando ayuda a los usuarios.

### **3.3.7. Reglas del Negocio.**

La modelación de un negocio, es la vía más eficiente de detectar sus problemas, entender su estructura y derivar los requerimientos del sistema a desarrollar. Para esto es imprescindible identificar los actores y trabajadores del negocio, en este caso, el negocio tiene los siguientes actores.

**Actores:**

- Usuario
- Administrador

**Casos de Uso:**

- Consulta de información.
- Realizar búsquedas en el catálogo de la biblioteca del Instituto.

**3.3.8. Modelo de Requerimientos****➤ Requisitos Funcionales.**

1. Poner a disposición de los usuarios documentos, libros, revistas, una hemeroteca, proyectos y tesis para su consulta a través de la web.
2. Funciones de búsqueda y visualización adecuada (por autor, por título, por año, por área).
3. Actualización y mantenimiento de la base de datos que contiene el catálogo de todos los recursos de la biblioteca, es decir permitirá: adicionar, eliminar, .modificar los datos de la base de datos de la biblioteca.
4. Diseñar una interfaz de usuario para ser presentado en web.
5. Diseño de una interfaz de usuario amigable y de fácil modificación.
6. Posibilidad de crear cuentas de suscripción de un usuario
7. Validar las cuentas de usuarios de los internautas.

## ➤ **Requisitos no Funcionales**

- **Apariencia o Interfaz Externa:**

El diseño de la interfaz deberá ser agradable y sobre todo lo más sencillo posible, pues será utilizada tanto por usuarios con una preparación integral como por algunos con conocimientos básicos de computación, por lo que además, deberá ser sencilla, aprovechando las facilidades del ambiente Web en el que se desarrollará.

- **Usabilidad:**

El sistema podrá ser usado por cualquier tipo de personas que posean conocimientos básicos en el manejo de la computadora y el ambiente Web en sentido general. Rendimiento: Aunque no se requiere una velocidad de respuesta comparada con los sistemas de tiempo real, se debe garantizar la rapidez de respuesta del sistema ante las solicitudes de los usuarios.

- **Portabilidad:**

El sistema podrá implantarse sobre Windows, Linux o Unix, de forma tal que no haya dificultad en cambiar de una a otra plataforma sin necesidad de efectuar cambios

significativos. Lo anterior se debe a que la aplicación está implementada sobre PHP que es un lenguaje multiplataforma.

- **Requerimientos de Software:**

Para la implantación del sistema se requiere de un servidor Unix, Linux o Windows, Apache HTTP Server como servidor Web, y MySQL 5.0 o superior como servidor de Bases de Datos. Los requerimientos en el lado del cliente para la utilización del sistema solo se limitan a tener disponible un navegador Web.

- **Requerimientos de Hardware:**

La computadora dedicada a servidor debe tener como mínimo las siguientes características de hardware: Procesador Dual Core, o similar, a 3.8 GHz, 2 Gb de memoria RAM y 250 Gb de capacidad en disco duro. Las computadoras que pueden ser utilizadas por los usuarios para acceder al sistema solo deben de estar conectadas en red con el servidor.



### **3.3.9. Método de Recopilación de Datos**

El método de recopilación de datos representa la estrategia concreta e integral de trabajo para el análisis de un problema o cuestión coherente con la definición teórica del mismo y con los objetivos de la investigación.

### **3.3.10. Método.**

Es el medio o camino a través del cual se establece la relación entre el investigador y el consultado para la recolección de datos. Y el método que usaremos para nuestra investigación es la encuesta.

#### **3.3.10.1. Encuesta.**

Consiste en obtener información de los sujetos de estudio, proporcionados por ellos mismos, sobre opiniones, conocimientos, actitudes o sugerencias.

#### **Ventajas:**

- Técnica de mayor utilización que permite obtener información de casi cualquier tipo de población.
- Estandarización: se harán las mismas preguntas a todos los elementos de la muestra apoyándonos en el cuestionario.

- Permite obtener información sobre hechos pasados de los encuestados.
- Gran capacidad para estandarizar datos, lo que permite su tratamiento informático y el análisis estadístico.
- Rapidez: en poco tiempo se puede reunir gran cantidad de información.
- Relativamente barata para la información que se obtiene con ello.

**Desventajas:**

- No permite analizar con profundidad temas complejos (recorrir a grupos de discusión).
- Dificultades para establecer relaciones causales.
- No toma en cuenta los factores contextuales que pueden interferir en las respuestas del sujeto.

**3.3.11. Instrumento.**

Es el mecanismo que utiliza el investigador para recolectar y registrar la información donde debe de poseer tres requisitos: Confiabilidad, Validez y Objetividad. Sin alguna de las tres condiciones, el instrumento no será útil, los resultados no serán legítimos y la investigación deberá ser rechazada.

### **3.3.11.1. Cuestionario.**

Es el método que utiliza un instrumento o formulario impreso, destinado a obtener respuestas sobre el problema en estudio y que el investigado o consultado llena por sí mismo.

#### **Pasos en la construcción del Instrumento de Recolección de Datos:**

- a. Listar las variables que se pretenden medir u observar.
- b. Revisar su definición conceptual y comprender su significado.
- c. Revisar cómo han sido definidas operacionalmente las variables.
- d. Elegir el instrumento o los instrumentos que hayan sido favorecidos por la comparación y adaptarlos al contexto de la investigación.
- e. Indicar el nivel de medición de cada variable
- f. Indicar como se van a codificar los datos.
- g. Aplicar una prueba piloto del instrumento de medición.
- h. Instrumento definitivo.

### **3.3.12. Lenguaje de Modelamiento Unificado**

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

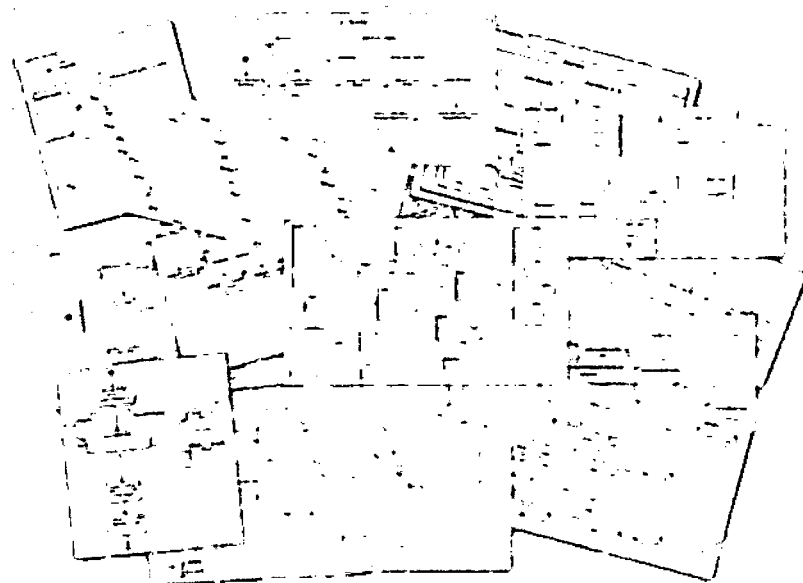
Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

FIGURA N° 09:

COLLAGE DE DIAGRAMAS UML.



### 3.3.13. Métrica de Validación de Software

ISO 9126 es un estándar internacional para la evaluación de la calidad del software. Está reemplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos. Este estándar es el más usado... El estándar está dividido en cuatro partes las cuales dirigen, realidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y sub características de la siguiente manera:

- **Funcionalidad.** Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas. Idoneidad - Atributos del software relacionados con la presencia y aptitud de un conjunto de funciones para tareas especificadas.
- **Exactitud.** Atributos del software relacionados con la disposición de resultados o efectos correctos o acordados.
- **Interoperabilidad.** Atributos del software que se relacionan con su habilidad para la interacción con sistemas especificados

- **Seguridad.** Atributos del software relacionados con su habilidad para prevenir acceso no autorizado ya sea accidental o deliberado, a programas y datos.

#### **Cumplimiento de normas.**

- **Fiabilidad.** Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.
- **Madurez.** Atributos del software que se relacionan con la frecuencia de falla por fallas en el software.
- **Recuperabilidad.** Atributos del software que se relacionan con la capacidad para restablecer su nivel de desempeño y recuperar los datos directamente afectados en caso de falla y en el tiempo y esfuerzo relacionado para ello.
- **Tolerancia a fallos.** Atributos del software que se relacionan con su habilidad para mantener un nivel especificado de desempeño en casos de fallas de software o de una infracción a su interfaz especificada.

- **Usabilidad.** Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
  
- **Aprendizaje.** Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
  
- **Comprensión.** Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
  
- **Operatividad.** Atributos del software que se relacionan con el esfuerzo de los usuarios para la operación y control del software.

**Atractividad.**

- **Eficiencia.** Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.



- **Comportamiento en el tiempo.** Atributos del software que se relacionan con los tiempos de respuesta y procesamiento y en las tasas de rendimientos en desempeñar su función.

### **Comportamiento de recursos**

- **Mantenibilidad.** Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.
- **Estabilidad.** Atributos del software relacionados con el riesgo de efectos inesperados por modificaciones.
- **Facilidad de análisis.** Atributos del software relacionados con el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallos, o identificaciones de partes a modificar.
- **Facilidad de cambio.** Atributos del software relacionados con el esfuerzo necesario para la modificación, corrección de falla, o cambio de ambiente.
- **Facilidad de pruebas.** Atributos del software relacionados con el esfuerzo necesario para validar el software modificado.

- **Portabilidad.** Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.
  
- **Capacidad de instalación.** Atributos del software relacionados con el esfuerzo necesario para instalar el software en un ambiente especificado.
  
- **Capacidad de reemplazamiento.** Atributos del software relacionados con la oportunidad y esfuerzo de usar el software en lugar de otro software especificado en el ambiente de dicho software especificado.
  
- **Adaptabilidad.** Atributos del software relacionados con la oportunidad para su adaptación a diferentes ambientes especificados sin aplicar otras acciones o medios que los proporcionados para este propósito por el software considerado.

### **Co-Existencia**

La sub característica Conformidad no está listada arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad.

Cada sub característica (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así. Como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

- Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).
- Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales.

Idealmente, la calidad interna no necesariamente implica calidad externa y esta a su vez la calidad en el uso.

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación.

## **CAPÍTULO IV**

### **RESULTADOS Y DISCUSIÓN**

#### **4.1. Definición de Requisitos del Sistema**

Esta actividad tuvo el objetivo de determinar los requisitos generales, mediante una serie de sesiones de trabajo con los usuarios participantes, los cuáles deben ser planificados y realizados. Una vez que las sesiones son finalizadas, se analizó la información obtenida definiendo los requisitos y sus prioridades, los cuáles se añaden al catálogo de requisitos que servirá para el estudio y valoración de las distintas alternativas de solución que se propongan.

#### **4.2. Identificación de Requisitos**

Para obtener las necesidades que ha de satisfacer el sistema actual se han decidido los tipos de sesiones de trabajos realizados y la frecuencia con que tienen lugar en función de la disponibilidad de los usuarios participantes.

También ha sido conveniente seleccionar la información de los sistemas

de información existentes que resultaron de interés para el desarrollo de dichas sesiones de trabajo, si ya se ha realizado el Estudio de la Situación Actual.

Después de establecidos los puntos anteriores, se planificaron las sesiones de trabajo con los usuarios participantes identificados al estudiar el alcance del Estudio de Viabilidad del Sistema, y se realizan de acuerdo al plan previsto. La información obtenida dependiendo del tipo de sesión de trabajo seleccionado.

#### **4.3. Catalogación de Requisitos**

Se realizó el análisis de la información obtenida en las sesiones de trabajo para la Identificación de Requisitos, definiendo y catalogando los requisitos (funcionales y no funcionales) que el sistema debe satisfacer indicando también sus prioridades. Además se incluyen los requisitos relativos a distribución geográfica y entorno tecnológico.

##### **Catálogo de Requisitos**

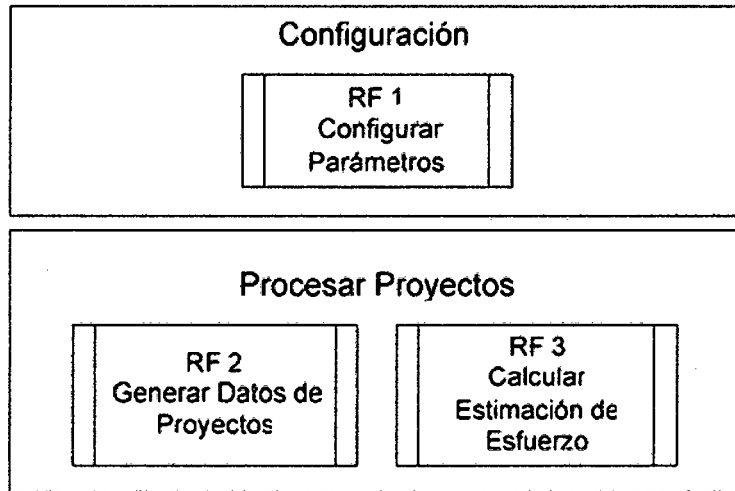
A continuación se describen los requisitos funcionales y no funcionales que fueron relevados para ser satisfechos por el sistema software. Se utiliza como guía de escritura el estándar IEEE para Especificación de los Requisitos del Software (ERS). (IEEE, 1998).

**a. Requisitos Funcionales:**

FIGURA N° 10:

REQUISITOS FUNCIONALES.

LOS PRINCIPALES REQUISITOS FUNCIONALES DEL SISTEMA:

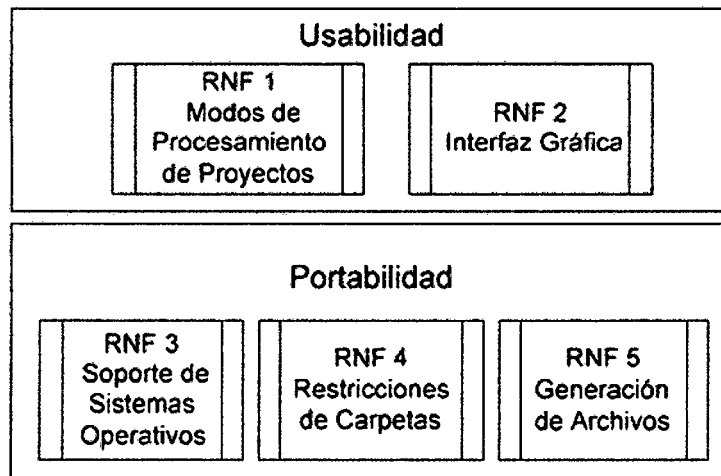


**b. Requisitos No Funcionales:**

FIGURA N° 11:

REQUISITOS NO FUNCIONALES.

SE MUESTRAN LOS REQUISITOS NO FUNCIONALES DEL SISTEMA:



#### **4.4. Estudio de Alternativas de Solución**

Esta actividad tiene el objetivo de proponer y estudiar alternativas de solución que respondan satisfactoriamente a los requisitos planteados, considerando también los resultados obtenidos en el Estudio de la Situación Actual (en el caso de que se haya realizado).

En la descripción de las distintas alternativas propuestas, se debe especificar si alguna de ellas se encuentra basada, total o parcialmente, en un producto existente en el mercado. Si la alternativa incluye un desarrollo a medida, se debe incorporar en la descripción de la misma un modelo abstracto de datos y un modelo de procesos, en orientación a objetos, un Modelo de Negocio y opcionalmente un Modelo de Dominio.

Además, considerando el ámbito y funcionalidad que debe cubrir el sistema, puede ser conveniente realizar una descomposición del sistema en subsistemas previamente a la definición de cada alternativa.

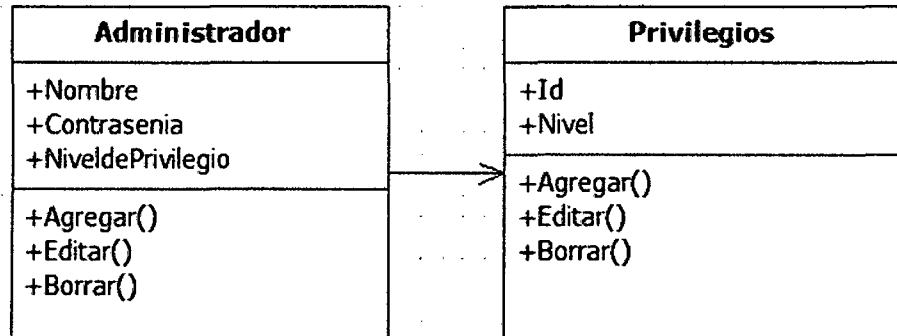


## 4.5. Modelamiento del Sistema

### 4.5.1. Diagrama de Clases

FIGURA N° 12:

DIAGRAMA DE CLASES



Clase Administrador mostrando el nivel de privilegios adquiridos a partir de la tabla Privilegios los cuales se adjudican en tiempo de diseño.

FIGURA N° 13:

DIAGRAMA DE CLASES

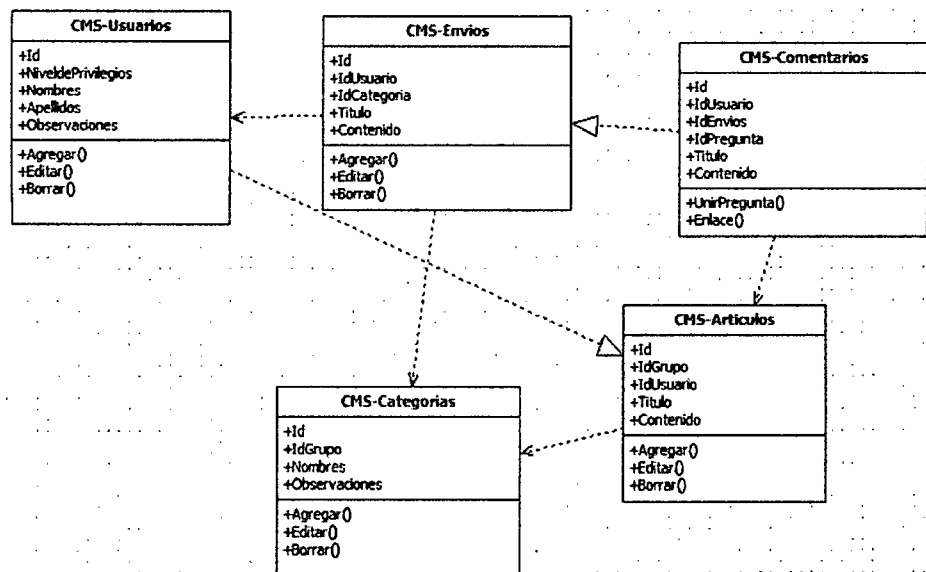


Diagrama de clases que muestra el nivel de dependencia de las clases que permiten administrar el CMS personalizado, donde se controla los envíos, noticias, artículos dependiendo del usuario que los envía incluso el administrador, pudiendo estos elementos ser modificados, eliminados.

Un CMS personalizado reducido en la menor cantidad de tablas reduciendo la complejidad para permitir enviar elementos, categorizarlos y por supuesto poder editarlos y borrarlos sin tener que afectar al resto de los envíos de los otros usuarios.

#### 4.5.2. Diagrama de Flujo

FIGURA N° 14.  
DIAGRAMA DE FLUJO

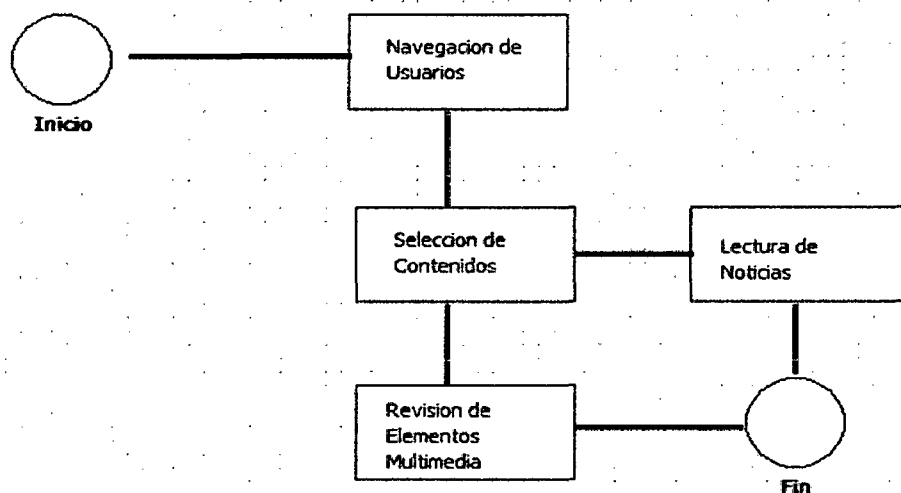
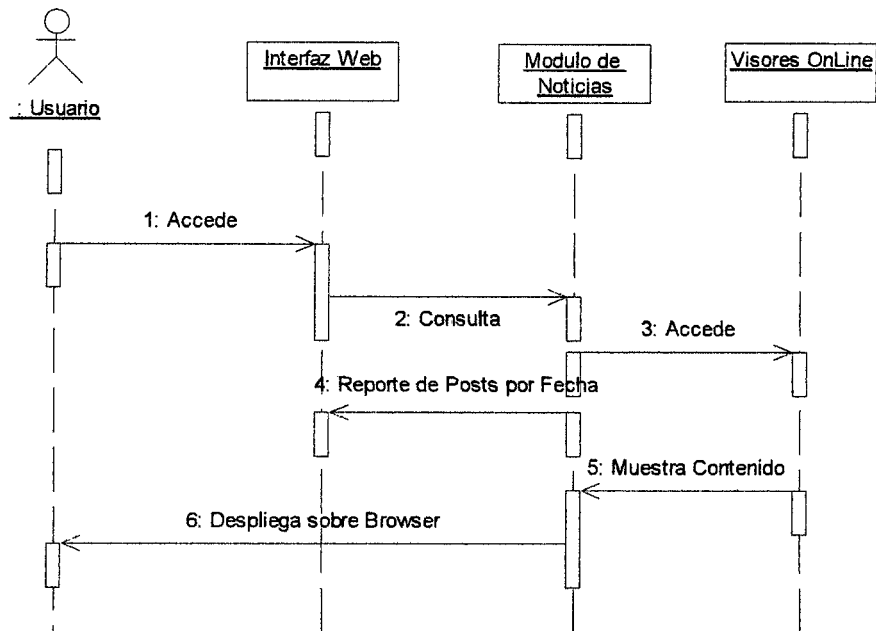


Diagrama de flujo de datos que muestra la secuencia de navegación en que los usuarios siguen durante la navegación y elección de acciones.

Se debe tener en cuenta que hay pocos elementos por que se desea reducir complejidad, simplificar procesos.

### 4.5.3. Diagrama de Secuencia

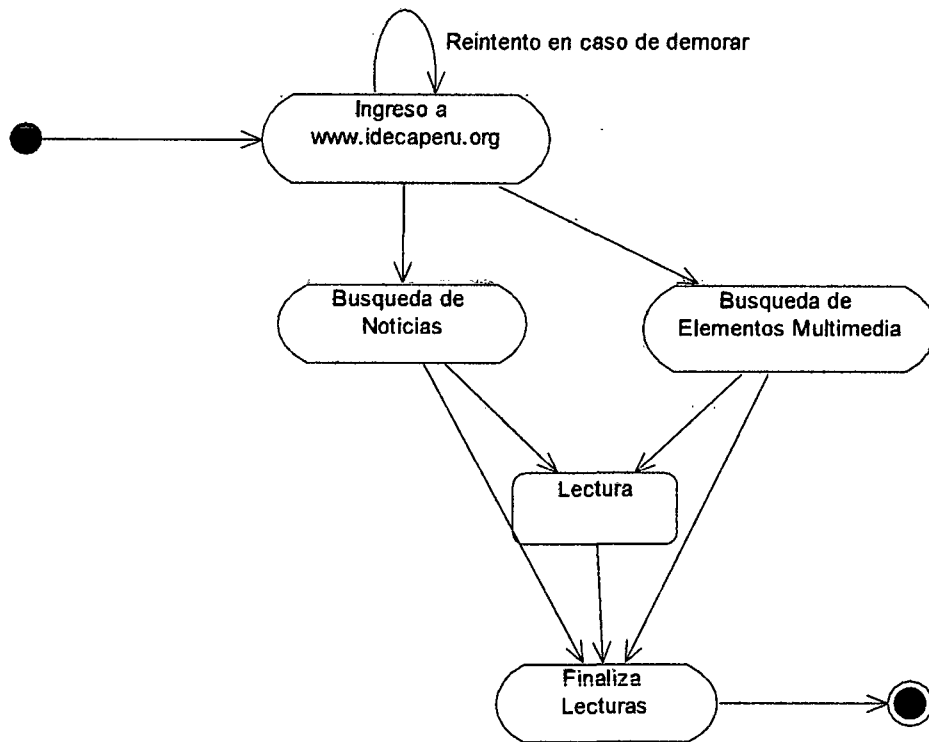
FIGURA N° 15:  
DIAGRAMA DE SECUENCIA



Los usuarios acceden a la Interfaz Web seguidamente realizan consultas sobre los modulos de noticias y envios (posts) este modulo a su vez les devuelve un reporte con los elementos que califican sobre los criterios de busqueda que pueden ser parte del titulo o contenido, asi como la fecha de envio, este mismo proceso se aplica a los contenidos multimedia, sobre los que se aplican criterios de busqueda pero solo basados en el titulo del video o libro PDF que se haya enviado al servidor.

#### 4.5.4. Diagrama de Estados

FIGURA N° 16:  
DIAGRAMA DE ESTADOS




Los usuarios acceden a la Interfaz Web en el caso de no acceder inmediatamente se reintenta el acceso a nuestra URL (Uniform Resource Locator) se realiza busquedas se cambia al estado de lectura de no acceder a este modulo se procede con la busqueda de multimedia o documentos PDF, se accede al estado de lectura y expectacion visual y finalmente termina el proceso.

## 4.6. Interfaces del Sistema

### a. Página Principal

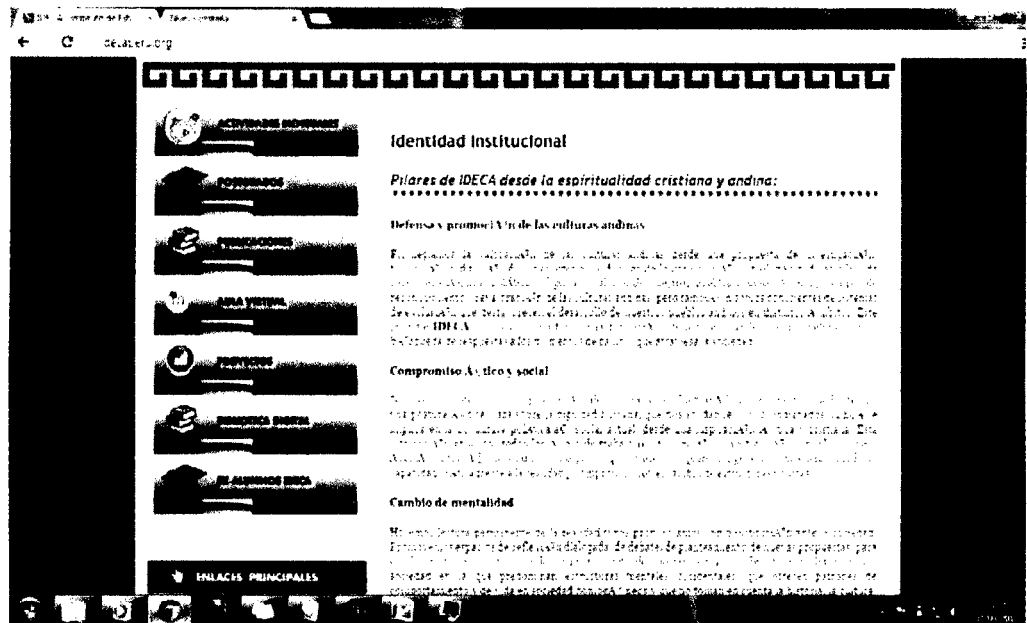
Aquí se muestra la plantilla principal sobre la que se administra el resto de elementos web de la página, a continuación de este grafico se muestra el código fuente de la plantilla CMS personalizada sin tener que duplicar el código fuente para ellos usando los esquemas de modularidad y reusabilidad.

FIGURA N° 17:  
PÁGINA PRINCIPAL

		
01	<?PHP	
02	//	
03	// Modulo index.php	
04	//	
05	include( "cms/header.php" );	
06	include( "cms/menu.php" );	
07	include( "cms/header.php" );	
08		
09	// contenido de este modulo	
10		
11	include( "cms/footer.php" );	
12	?>	
13		

## b. Módulo de Publicación de Noticias

FIGURA N° 18:  
PORTADA PRINCIPAL



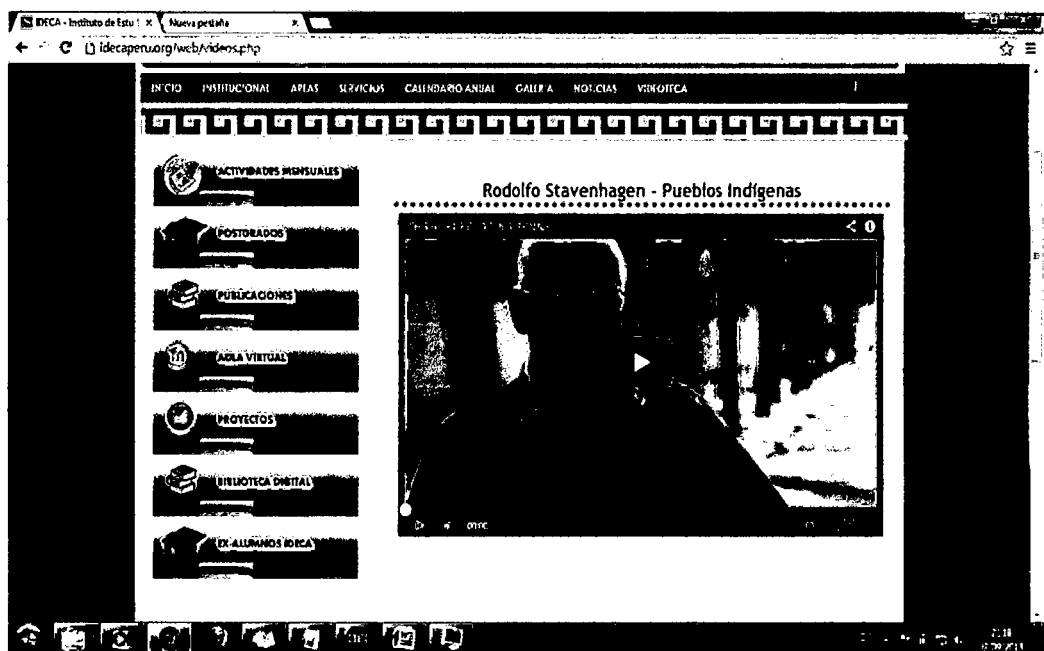
En este módulo se muestran como contenido web lo que previamente se ha almacenado en la base de datos agregándoles formatos de alineación y estilos se muestra el título en contraste a comparación del contenido sobre los marcadores. **<P> contenido </P>**.

## c. Módulo de Visores OnLine

Permite mostrar un contenido multimedia o de documentos PDF y/o Word sobre la misma página web sin tener que cargar o entrar a la pugna donde está la fuente de dicho contenido multimedia.

Un claro ejemplo se muestra a continuación al mostrar un video almacenado en el servidor de video youtube.com pero para ver el contenido no se requiere cerrar la página web actual sino que carga el contenido en un módulo integrado a nuestra página web. Un módulo incrustado como visor.

FIGURA N° 19:  
MÓDULO DE VISORES ONLINE



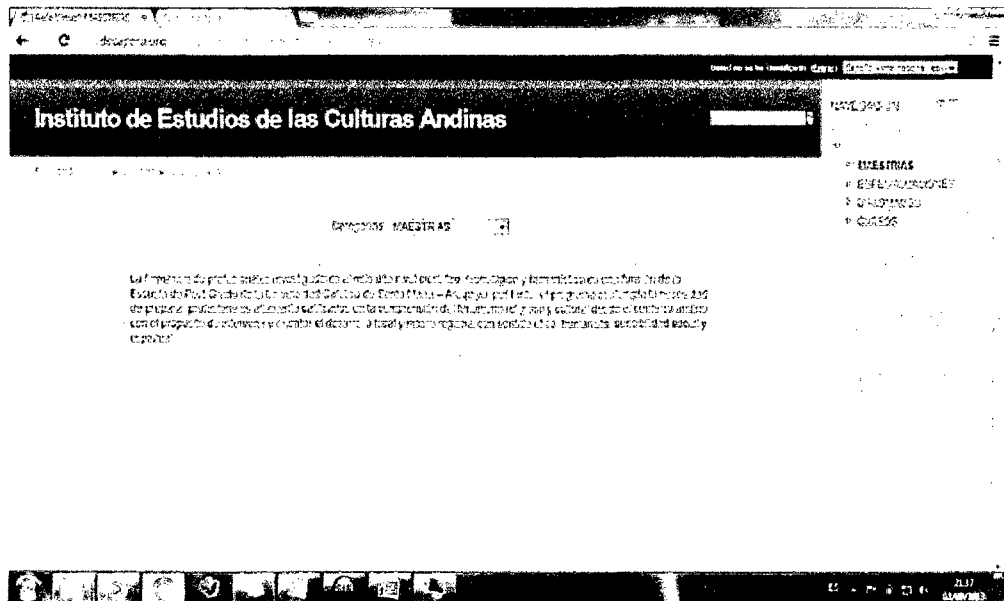
Siguiendo este planteamiento puede mostrarse documentos PDF o WORD sobre la página web sin tener que salir de página web o actualizar el contenido del browser para mostrar los elementos multimedia, sino que se muestran en un módulo incrustado basado en un `<iframe></iframe>`.

#### d. Módulo de Aulas Virtuales

Aquí no se ha implementado el código fuente para estas aulas virtuales sino que se ha hecho uso de Moodle que viene en el servidor adquirido al registrar el dominio **www.idecaperu.org** directamente sobre el Panel de Control.

La instalación del módulo **Moodle** se hace sobre la carpeta **/public\_html/aulavirtual**.

FIGURA N° 20:  
MÓDULO DE VISORES ONLINE





## **CONCLUSIONES**

Se Analizó y Diseño el CMS para poder adaptar los servicios de la institución y adaptarlos al contenido visible de la Pagina Web, ofreciendo así la información necesaria además de ofrecer servicios de publicación cultural y actividades que IDECA realiza como parte de su proyección hacia la sociedad.

Se ha Implementado la Base de Datos para el Sistema Web, tomando los criterios de Bases de Datos con respecto a las 4 primeras formas normales, siendo las más requeridas para la implementación del administrador CMS.

Se ha Implementado Visores On-Line para documentos y videos que han sido alojados previamente en el servidor de videos de **youtube**, creando así un enlace directo entre la página web de IDECA y el video reproducido sin salir del sitio WEB así como el visor de documentos PDF sin la necesidad de Plugins externos.

Se ha desarrollado la Interfaz de Administrador de sistema para que se pueda administrar los contenidos, sin tener que preocuparse por el contenido HTML sino directamente editarlos como un procesador de textos, de modo que incluso pueda agregar gráficos y tablas de forma rápida sencilla.

## BIBLIOGRAFÍA

- Fernández Velásquez Ángel (1998), Metodología de la Investigación Científica, Editorial San Marcos, Lima Perú.
- Allison Jack, (2011), La Biblia de MySQL y Apache para desarrolladores Web y CMS.
- Albrecht, A. (1983), Software Function, Source Lines of Code and Development Effort Prediction.
- Bielak, J. (2000). Improving Size Estimates Using Historical Data.
- Britos, P. (2008). Procesos de Explotación de Información Basados en Sistemas Inteligentes. Tesis Doctoral. Facultad de Informática. Universidad Nacional de la Plata.
- DAMIÁN, P. (2007). Que son las bases de datos.  
<http://www.maestrosdelweb.com/>
- Miguel, A. (2001). ¿Qué es PHP?  
<http://www.desarrolloweb.com/articulos/392.php>
- Definición de JavaScript  
[http://www.librosweb.es/javascript/pdf/introduccion\\_javascript.pdf](http://www.librosweb.es/javascript/pdf/introduccion_javascript.pdf)

- Browser, explorador, navegador web.  
<http://es.scribd.com/doc/33556783/Browser-Definicion-y-Tipos>
- Métricas de calidad.  
<http://ldc.usb.ve/~abianc/materias/ci4712/metricas.pdf>
- La norma ISO/IEC 9126.  
<http://iso25000.com/index.php/iso-iec-9126.html>
- Administración de la cobranza.  
<http://www.buenastareas.com/ensayos/Administraci%C3%B3n-De-La-Cobranza/170004.html>
- Definición de Control.  
<http://www.monografias.com/trabajos14/control/control.shtml#def>
- Cuenta de usuario.  
[http://es.wikipedia.org/wiki/Cuenta\\_de\\_usuario](http://es.wikipedia.org/wiki/Cuenta_de_usuario)
- El control como fase del proceso administrativo.  
<http://www.gestiopolis.com/recursos/documentos/fulldocs/ger1/controladm.html>
- Interfaz de usuario.  
[http://es.wikipedia.org/wiki/Interfaz\\_de\\_usuario](http://es.wikipedia.org/wiki/Interfaz_de_usuario)
- La cobranza  
<http://www.primeraexportacion.com.ar/documentos-tecnicos/242-cobranza.html>.

## **ANEXOS**

## **ANEXO – I**

**FICHA DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO ESTÁNDAR ISO-9126**

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
<b>1. FUNCIONALIDAD</b>					
<b>Adecuación:</b> la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					
<b>Exactitud:</b> la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.					
<b>Interoperabilidad:</b> la capacidad del producto software para interactuar con uno o más sistemas especificados					
<b>Seguridad:</b> referido a la capacidad del producto software para proteger la información y los datos.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad.					
<b>2. FIABILIDAD</b>					
<b>Madurez:</b> la capacidad del producto software para evitar fallos provocados por errores en el software.					
<b>Tolerancia a fallos:</b> la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.					
<b>Recuperabilidad:</b> la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.					

<b>3. USABILIDAD</b>					
<b>Comprensibilidad:</b> la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.					
<b>Facilidad de aprendizaje:</b> la capacidad del producto software para permitir al usuario aprender su aplicación.					
<b>Atracción:</b> la capacidad del producto software para atraer al usuario.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.					
<b>Operabilidad:</b> la capacidad del producto software para permitir que el usuario lo opere y lo controle.					
<b>4. EFICIENCIA</b>					
<b>Comportamiento temporal:</b> la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.					
<b>Utilización de recursos:</b> la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					
<b>5. MANTENIBILIDAD</b>					
<b>Analizabilidad:</b> Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.					
<b>Cambiabilidad:</b> Capacidad del producto software de permitir implementar una modificación especificada. La					



implementación incluye los cambios en el diseño, el código y la documentación.					
<b>Estabilidad:</b> Capacidad del producto software de evitar los efectos inesperados de las modificaciones.					
<b>Facilidad de prueba:</b> Capacidad del producto software de permitir validar las partes modificadas.					
<b>Conformidad:</b> Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.					
<b>INDICADORES</b>	<b>PUNTUACIÓN</b>				
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>6. PORTABILIDAD</b>					
<b>Adaptabilidad:</b> la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.					
<b>Facilidad de instalación:</b> la capacidad del producto software para ser instalado en un ambiente determinado.					
<b>Coexistencia:</b> la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.					
<b>Reemplazabilidad:</b> la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.					
<b>SUB TOTALES</b>					
<b>TOTAL</b>					

Fuente: Ficha de evaluación ISO-9126

**FICHA DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO ESTÁNDAR ISO-9126**

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
<b>1. FUNCIONALIDAD</b>					
<b>Adecuación:</b> la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					
<b>Exactitud:</b> la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.					
<b>Interoperabilidad:</b> la capacidad del producto software para interactuar con uno o más sistemas especificados					
<b>Seguridad:</b> referido a la capacidad del producto software para proteger la información y los datos.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad.					
<b>2. FIABILIDAD</b>					
<b>Madurez:</b> la capacidad del producto software para evitar fallos provocados por errores en el software.					
<b>Tolerancia a fallos:</b> la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.					
<b>Recuperabilidad:</b> la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.					

<b>3. USABILIDAD</b>					
<b>Comprensibilidad:</b> la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.					
<b>Facilidad de aprendizaje:</b> la capacidad del producto software para permitir al usuario aprender su aplicación.					
<b>Atracción:</b> la capacidad del producto software para atraer al usuario.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.					
<b>Operabilidad:</b> la capacidad del producto software para permitir que el usuario lo opere y lo controle.					
<b>4. EFICIENCIA</b>					
<b>Comportamiento temporal:</b> la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.					
<b>Utilización de recursos:</b> la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					
<b>5. MANTENIBILIDAD</b>					
<b>Analizabilidad:</b> Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.					
<b>Cambiabilidad:</b> Capacidad del producto software de permitir implementar una modificación especificada. La					

implementación incluye los cambios en el diseño, el código y la documentación.					
<b>Estabilidad:</b> Capacidad del producto software de evitar los efectos inesperados de las modificaciones.					
<b>Facilidad de prueba:</b> Capacidad del producto software de permitir validar las partes modificadas.					
<b>Conformidad:</b> Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.					
<b>INDICADORES</b>	<b>PUNTUACIÓN</b>				
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>6. PORTABILIDAD</b>					
<b>Adaptabilidad:</b> la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.					
<b>Facilidad de instalación:</b> la capacidad del producto software para ser instalado en un ambiente determinado.					
<b>Coexistencia:</b> la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.					
<b>Reemplazabilidad:</b> la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.					
<b>Conformidad:</b> la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.					
<b>SUB TOTALES</b>					
<b>TOTAL</b>					

Fuente: Ficha de evaluación ISO-9126

CUADRO N<sup>o</sup> 02  
 ESCALA VALORATIVA. (ESCALA DE LIKERT)

<b>Indicador Cualitativo</b>	<b>Valor</b>
Deficiente	1
Malo	2
Regular	3
Bueno	4
Muy bueno	5

Fuente: Escala valorativa. (Escala de Likert)

CUADRO N<sup>o</sup> 03  
 CUADRO DE DECISIONES ISO 9126.

<b>Clasificación</b>	<b>Intervalo</b>	<b>Decisión</b>
A) Inaceptable	[ 27 - 54 >	
B) Mínimamente aceptable	[ 54 - 81 >	
C) Aceptable	[ 81 - 95 >	
D) Cumple los requisitos	[ 95 - 122 >	
E) Excede los requisitos	[ 122 - 135 ]	

Fuente: Cuadro de decisiones ISO 9126.

## **ANEXO – II**

## FICHAS DE EVALUACIÓN REALIZADAS A LOS INSCRIPTORES ISO – 9126.

INDICADORES	PUNTAJE INSCRIPTOR - 1				PUNTAJE INSCRIPTOR - 2				PUNTAJE INSCRIPTOR - 3				PUNTAJE INSCRIPTOR - 4				PUNTAJE INSCRIPTOR - 5				PUNTAJE INSCRIPTOR - 6				PUNTAJE INSCRIPTOR - 7											
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
<b>1. FUNCIONALIDAD</b>																																				
<b>Adecuación:</b> la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.																																				
<b>Exactitud:</b> la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.																																				
<b>Interoperabilidad:</b> la capacidad del producto software para interactuar con uno o más sistemas especificados.																																				
<b>Seguridad:</b> referido a la capacidad del producto software para proteger la información y los datos.																																				
<b>Conformidad:</b> la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad relativos a la funcionalidad.																																				
<b>2. FIABILIDAD</b>																																				
<b>Madurez:</b> la capacidad del producto software para evitar fallos provocados por errores en el software.																																				
<b>Tolerancia a fallos:</b> la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.																																				
<b>Recuperabilidad:</b> la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.																																				







CUADRO N° 04

ESCALA VALORATIVA. (ESCALA DE LIKERT)

Indicador Cualitativo	Valor
Deficiente	1
Malo	2
Regular	3
Bueno	4
Muy bueno	5

Fuente: Escala valorativa ISO-9126

CUADRO N° 05

CUADRO DE DECISIONES ISO - 9126

Clasificación	Intervalo	Decisión 1	Decisión 2	Decisión 3	Decisión 4	Decisión 5	Decisión 6	Decisión 7
A) Inaceptable	[ 27 - 54 >							
B) Mínimamente aceptable	[ 54 - 81 >							
C) Aceptable	[ 81 - 95 >							
D) Cumple los requisitos	[ 95 - 122 >	103	102	108	100	98	107	99
E) Excede los requisitos	[ 122 - 135 ]							

Fuente: Cuadro de decisiones ISO - 9126

## **ANEXO - III**

# UNIVERSIDAD NACIONAL DEL ALTIPLANO

## ESCUELA DE POST GRADO EN INGENIERÍA INFORMÁTICA

### MANUAL DE USUARIO

1. Acceder al sitio WEB <http://idecaperu.org/>

FIGURA N° 21:  
INSTITUTO DE ESTUDIOS DE LAS CULTURAS ANDINAS



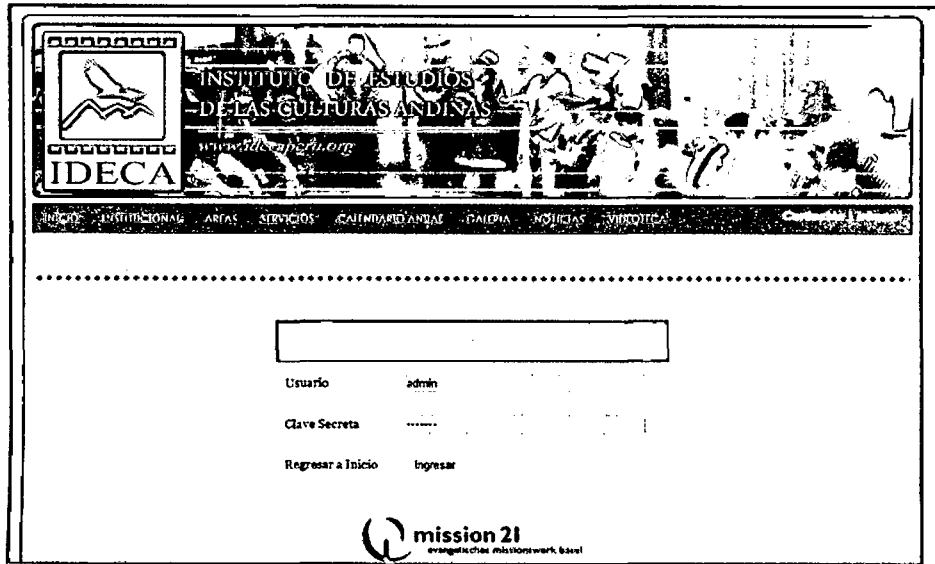
2. Para acceder al Panel de Administrador

FIGURA N° 22:  
PANEL DE ADMINISTRADOR



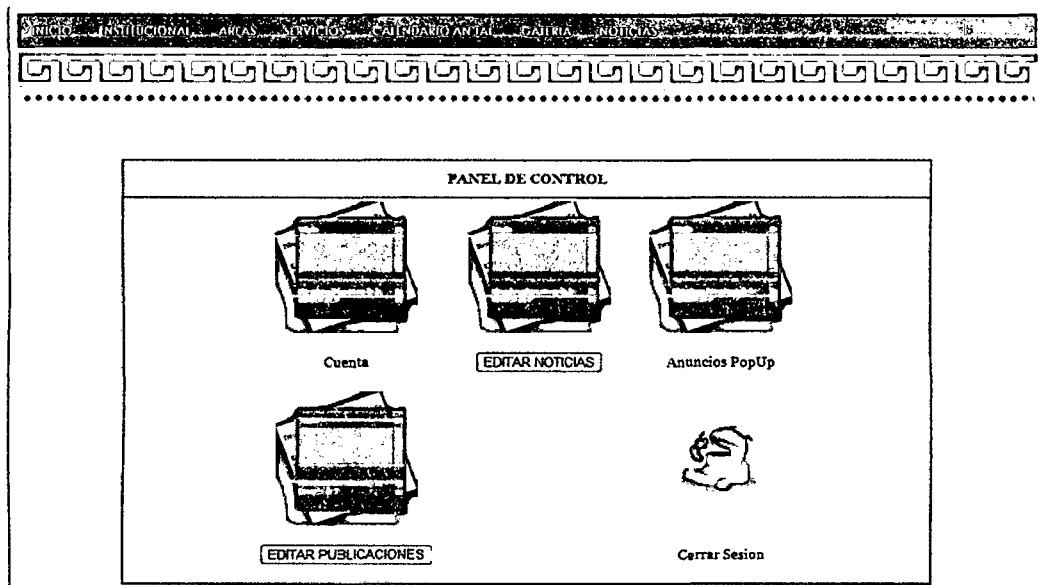
Deberá hacer click sobre el enlace Intranet

FIGURA N° 23:  
VALIDACION DE CONTRASEÑAS



Una vez aquí, se ingresa la clave de administrador y prosigue la validación desde el Panel de Control, listándose los elementos de edición de noticias y artículos.

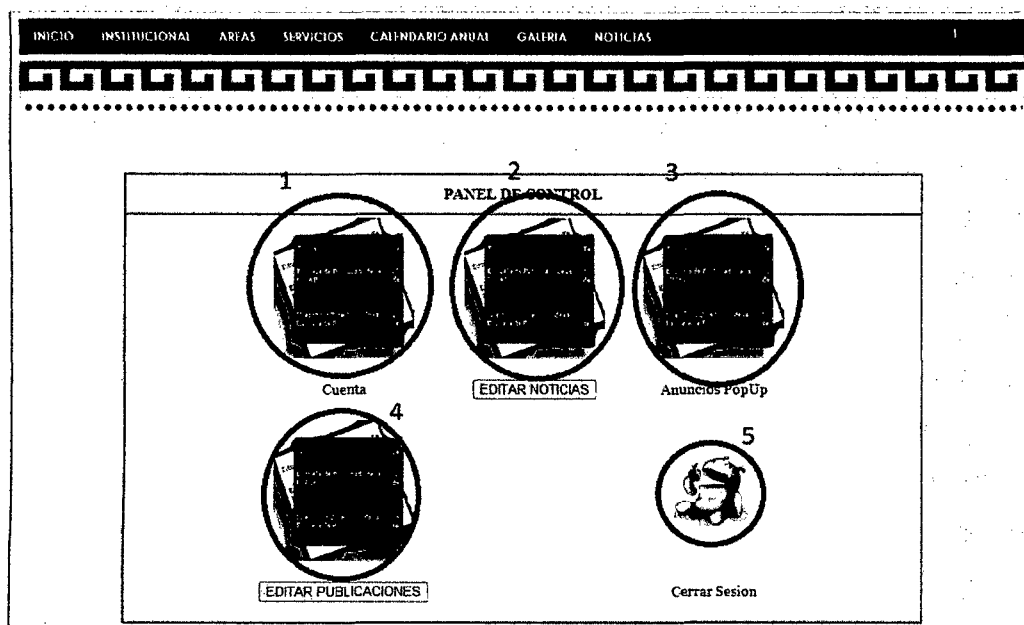
FIGURA N° 24:  
VISTA DEL PANEL DE CONTROL



A Continuación detallamos las partes del Panel de Control describiendo los elementos que permiten administrar los elementos de nuestro CMS.

FIGURA N° 25:

DESCRIBIENDO EL PANEL DE CONTROL



1. Mantenimiento de Cuentas de usuario.
2. Mantenimiento de Noticias y Artículos.
3. Administración de Elementos emergentes o mensajes PopUp.
4. Administración de Publicaciones (aun implementándose)
5. Cerrar de la Sesión

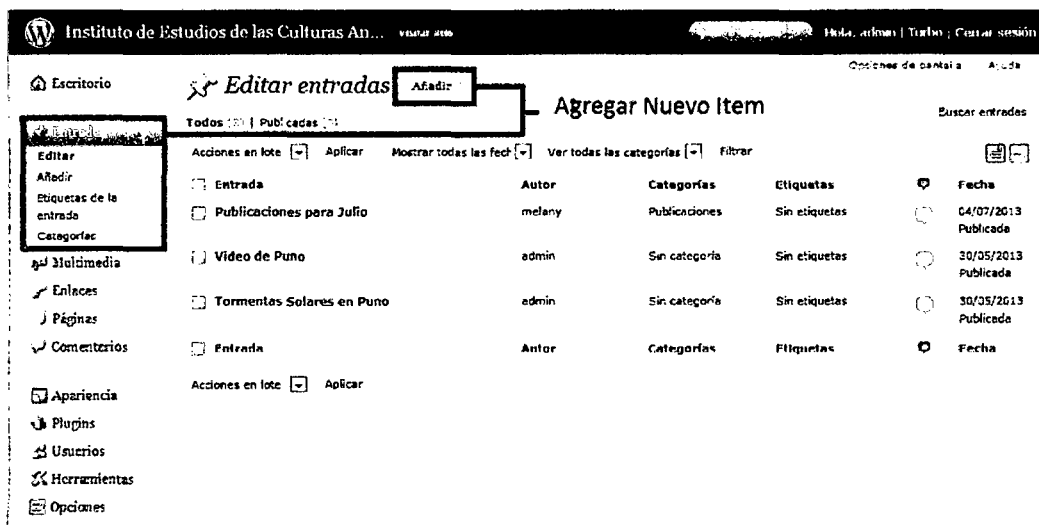
DESCRIPCION:

1. Mantenimiento de Cuentas de usuario.

Contiene las opciones básicas: **Agregar / Editar / Eliminar.**

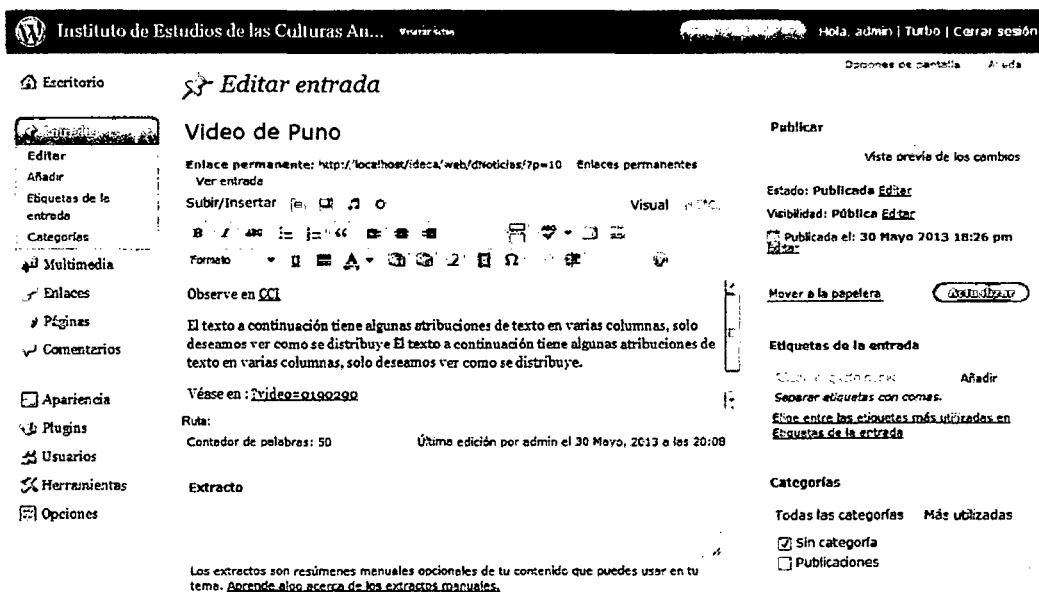
2. Mantenimiento de Noticias y Artículos.

FIGURA N° 26:  
EDITAR ENTRADA



A continuación podemos Editar o crear un nuevo contenido y mientras se edita tener el acabado visual para poder editarlo como si fuera un editor de texto normal de escritorio.


FIGURA N° 27:  
EDITAR ENTRADA



### 3. Administración de Elementos emergentes o mensajes PopUp.

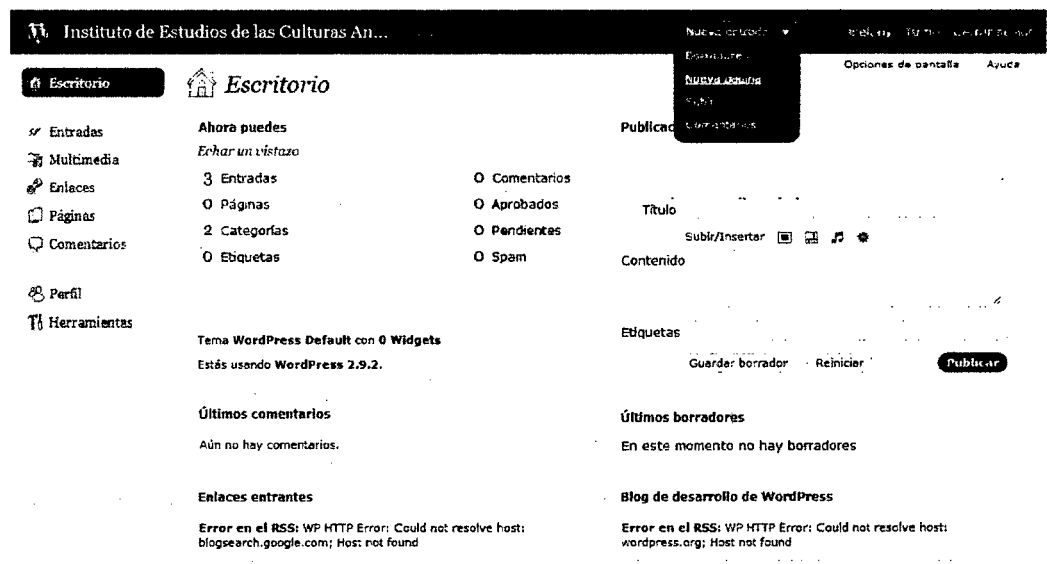
Aquí tenemos las opciones para agregar nuevos elementos al PopUp, editarlos y por supuesto poder eliminarlos, se muestra únicamente el último elemento.

FIGURA N° 28:  
CONTROL DE ANUNCIOS

CONTROL DE ANUNCIOS		
Imagen	Referencia	Opciones
	<a href="http://www.google.com">http://www.google.com</a>	Nuevo Editar Borrar

### 4. Administración de Publicaciones

FIGURA N° 29:  
ESCRITORIO



The screenshot shows the WordPress dashboard interface. At the top, there is a navigation bar with the site name 'Instituto de Estudios de las Culturas An...' and a user profile dropdown menu. The main content area is divided into several sections:

- Left Sidebar:** A vertical menu with icons for 'Escritorio', 'Entradas', 'Multimedia', 'Enlaces', 'Páginas', 'Comentarios', 'Perfil', and 'Herramientas'.
- Top Center:** A section titled 'Ahora puedes' (Now you can) with a sub-section 'Echar un vistazo' (Take a look) showing statistics: 3 Entradas, 0 Páginas, 2 Categorías, and 0 Etiquetas.
- Top Right:** A 'Publicar' (Publish) button with a dropdown menu containing 'Nueva publicación', 'Escribir', 'Nueva etiqueta', 'Subir', and 'Comentarios'.
- Bottom Left:** A section for 'Tema WordPress Default con 0 Widgets' and 'Estás usando WordPress 2.9.2'. Below it, 'Últimos comentarios' (Latest comments) shows 'Aún no hay comentarios.' (No comments yet). 'Enlaces entrantes' (Incoming links) shows an error: 'Error en el RSS: WP HTTP Error: Could not resolve host: blogsearch.google.com; Host: not found'.
- Bottom Right:** A section for 'Últimos borradores' (Latest drafts) showing 'En este momento no hay borradores.' (No drafts at the moment). Below it, 'Blog de desarrollo de WordPress' (WordPress development blog) shows an error: 'Error en el RSS: WP HTTP Error: Could not resolve host: wordpress.org; Host not found'.



5. Cerrar de la Sesión

FIGURA N° 30:  
VALIDACION DE USUARIOS



Cerrar Sesion

Validación de Usuarios	
Usuario	<input type="text"/>
Clave Secreta	<input type="password"/>
Regresar a Inicio	<input type="button" value="Ingresar"/>