

872748
C



Universidad Don Vasco, A.C.

--- INCORPORACIÓN No. 8727-48 ---
a la Universidad Nacional Autónoma de México
Escuela de Informática

Implantación del sistema de portales
multidinámicos, COL 1.0 en el
Instituto SECOMEX, A.C.

Se deposita en las bibliotecas de la Universidad Don Vasco, para su depósito en formato electrónico e impresión.

Edición: 2007
Eduardo Alonso Méndez Rodríguez
P. N. Escamilla

TESIS CON
FALLA DE ORIGEN

TESIS:

Que para obtener el título de:

LICENCIADO EN INFORMÁTICA

Presenta:

Eduardo Alonso Méndez Rodríguez

Asesor:

Ing. Martha Catalina Núñez Escamilla



Uruapan, Michoacán, Mayo del 2003



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINACIÓN DISCONTINUA

Agradecimientos

**A Dios, por haberme dado esta oportunidad
y haberme dado la fuerza para aprovecharla...**

**A mis padres, porque ellos me han formado
y de ellos es este resultado por el que siempre
lucharon y se preocuparon...**

**A mis profesores, por haberme enseñado a aprender,
resolver los problemas y enfrentar la vida con decisión...**

**A mis amigos, por disfrutar de este logro junto a mi
y haberme dejado compartirlo con ellos...**

**Cynthia, porque siempre hiciste que mis problemas fueran menos
y nunca dejaste de preocuparte por mi,
haciendo de este trabajo tu propio proyecto...**

TESIS CON
FALLA DE ORIGEN

INDICE GENERAL

Introducción.....	1
Capítulo 1.- Software Libre	
1.1.- ¿Que es la informática?	2
1.2.- El Hardware en general.	3
1.3.- El Software en general.	4
1.4.- Tipos de software.....	6
1.5- Inicio y Razón del FreeSoftware (Software Libre).....	6
Capítulo 2.- Internet	
2.1.- TCP/IP.	22
2.2.- SLIP/PPP.	24
2.3.- DNS.	24
2.4.- World Wide Web	25
2.4.1.- Los URIS	26
2.4.2.- Los protocolos.....	27
2.4.3.- HTML.....	28
2.4.4.- Anclas.....	29
Capítulo 3.- Páginas WEB.	
3.1.- Como funciona la WEB.	31
3.2.- Antecedentes de la WEB.	33
3.3.- Estructura de una pagina WEB.....	34
Capítulo 4.- SQL & mySQL.	
4.1.- ¿Qué es SQL?.....	37
4.2.- ¿Qué es mySQL?.....	38
4.3.- ¿Porqué usar mySQL?.....	39
4.4.- Breve historia mySQL.....	40
4.5.- Principales características.	41
4.6.- Sistemas operativos soportados por MySql.....	42

Capítulo 5.- UNIX

5.1.- Historia de Unix.	44
5.2.- Características Generales.....	46
5.3.- El núcleo del sistema operativo.....	47
5.4.- Linux.	47
5.4.1.- El inicio.....	48

Capítulo 6.- HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS

6.1.- Métodos para el desarrollo de sistemas	51
6.1.1 Método del desarrollo del Análisis Estructurado.....	51
6.1.2 Ciclo de Vida del Desarrollo de Sistemas.....	53
6.2 Modelos para el desarrollo de sistemas	54
6.2.1 Modelo Evolutivo.....	55

Capítulo 7.- Análisis del Instituto SECOMEX A. C.

7.1.- Levantamiento de información del Instituto SECOMEX A. C.....	58
7.2.- Plan de proyecto de software.....	59
7.3.- Recursos del proyecto	61
7.4.- Diagrama de Contexto.....	63
7.5.- Diccionario de Datos.....	63
7.6.- Diagrama de Nivel 0.....	64
7.7.- Diagramas de Nivel 1	65
7.8.- Diseño Arquitectónico.....	80
7.10.- Diseño de las pantallas de alumnos.....	82
7.11.- Diálogos en línea.....	86
7.11.1 Ayuda.....	87
7.12.- Estructura de módulos y datos globales.....	87
7.13.- Funciones dentro de cada formulario.....	89
7.13.1.- Variables globales y finalidad.....	90

TESIS CON
FALLA DE ORIGEN

Capítulo 8.- Implantación del Sistema COL 1.0 en el Instituto SECOMEX A. C

8.1 Sistema Operativo e instalación de módulos necesarios	91
8.2 Manejo y funcionamiento del sistema	97

Capítulo 9.- Mantenimiento del Sistema COL 1.0

9.1 Detección de errores en COL 1.0	102
---	-----

Conclusiones.	108
----------------------------	------------

Bibliografía	109
---------------------------	------------

E

TESIS CON FALLA DE ORIGEN

INTRODUCCIÓN

Implantación del Sistema COL (Control On Line) en el Instituto SECOMEX A. C.

Al utilizar sistema bajo licencia GPL (software libre) se cuenta con una serie de soluciones que, además de ya haber sido implantadas en algún lugar y contar con documentación acerca de su desarrollo y funcionamiento, pueden no funcionar del todo bien al adecuarse en otro ambiente, además de tener la responsabilidad de mantener ese sistema como si nosotros mismos lo hubiéramos desarrollado. Por esto es fundamental contar con bases fuertes en diseño, desarrollo y programación de sistemas para poder entender ideas que no son de nuestra propia inspiración, pero que de una u otra forma son una solución real para un posible problema dentro de una empresa.

Cualquier empresa sea cual sea su giro debe de tener presencia en la internet, así como brindar servicios a sus usuarios y empleados desde cualquier punto del planeta. Una de las vías para encontrar soluciones es el mundo del GNU (Gnu is Not Unix) el cual nos brinda un sin fin de aplicaciones completamente desarrolladas para cualquier tipo de empresa, dando la posibilidad de modificar en cualquier momento su funcionamiento sin caer en ninguna repercusión legal, dado su naturaleza GPL (GNU Public Licence). Uno de estos millares de sistemas es PHP-Nuke, el cual es un sistema de portales Web, que cuenta con un diseño y mantenimiento 100 % profesional alrededor del mundo y nos

TESIS CON
FALLA DE ORIGEN

dada su estructura nos permite modificar su código para adaptarlo al funcionamiento de un Control escolar Vía Intranet-Internet.

El Objetivo General es implantar el sistema COL (Control On Line) para el Instituto SECOMEX A. C.

Como Objetivos Particulares tenemos los siguientes:

- a) Dar una imagen a la institución en el Internet (servicios que ofrece, historia de la empresa, objetivos, ubicación, dirección etc).
- b) Instalar un servicio de noticias para los visitantes con tópicos de interés informático.
- c) Instalar un Servicio para el control de las calificaciones de los alumnos, en donde el profesor podrá asignar las calificaciones correspondientes a sus alumnos y solo el alumno afectado pueda revisar esta información, todo esto desde cualquier parte del planeta.
- e) Dar mantenimiento a los sistemas detectando y corrigiendo los fallos y anomalías que se puedan presentar.

TESIS CON
FALLA DE ORIGEN

Nuestra hipótesis señala que los portales multidinámicos ofrecen servicios que facilitan el acceso a información, ahorrando tiempo, dinero y esfuerzo. Los portales multidinámicos resolverán el problema de las distancias. Los portales multidinámicos pueden dar solución a los problemas de control-administración, descentralizando así el lugar operacional, ampliándose a cualquier parte del planeta donde se cuente con conexión de internet, logrando así el incremento de beneficios a los usuarios y clientes, por lo tanto

umentando el beneficio propio de la empresa o institución. Se utilizará un portal multidinámico con funciones de control escolar corrigiendo el comportamiento de este para lograr su funcionamiento dentro del instituto cubriendo las expectativas.

El desarrollo de aplicaciones y sistemas se ha revolucionado, movimientos relativamente nuevos han cobrado mucha fuerza y amenazan con romper los viejos esquemas económicos, así como la forma de trabajar y desarrollar soluciones. Un esquema totalmente fuera de lo común, está invadiendo el mundo de la informática y es imposible no tomarlo en cuenta. Este "fenómeno" ha capturado la atención de las empresas informáticas mas importantes del planeta, desde IBM, hasta el gigante Microsoft. Y lo mejor de todo es que esta al alcance de cualquier usuario de computadoras, de cualquier programador.

Se tratará de visualizar la importancia de este movimiento que sin otros procesos alternos a el no hubiera sido posible, así como demostrar como un modelo tan diferente a los conocidos puede traernos muchos beneficios, tanto económicos, como de aprendizaje.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO I

EL SOFTWARE LIBRE

El software libre se ha caracterizado como un fenómeno dentro de la informática rompiendo con muchos esquemas y estándares económicos y de trabajo, compitiendo con empresas grandes y establecidas, aportando avances importantes en cuanto a la forma de pensar y desarrollar sistemas.

El software libre es mas que una filosofía o una idea, es ya una realidad, y sus avances han puesto a temblar a varios gigantes de esta importante área. El software libre abarca desde soluciones para niños, hasta llegar a soluciones científicas avanzadas sin dejar de mencionar los juegos en sus varias facetas.

1.1.-¿Qué es la informática?

La informática es la técnica utilizada para almacenar, procesar y distribuir la información de una forma automática. En la actualidad la informática se auxilia de medios electrónicos para lograr este fin. La informática se auxilia de varias técnicas para el tratado de información, y estas técnicas van evolucionando día a día, siendo una de las áreas que más cambios y avances presenta a un ritmo realmente vertiginoso. La informática se divide en dos partes esenciales: *software* y *hardware*, los cuales dan forma a todo este grupo de tecnología.

TESIS CON
FALLA DE ORIGEN

1.2.- El Hardware en general.

El hardware de la computadora se refiere a los elementos físicos que la conforman: las piezas que podemos tocar. Por ejemplo, la mayoría de las computadoras tienen un teclado con el que se escribe y un monitor o pantalla donde se visualiza la salida de la computadora. La caja donde se encuentra alojada la computadora (su gabinete) contiene otros elementos de hardware, como pueden ser uno o más discos duros que le permiten almacenar datos.

Otro componente es la memoria, también llamada RAM (*Random Access Memory, Memoria de acceso aleatorio*). La memoria es el espacio de trabajo de la computadora que contiene el código, los datos y demás información de los programas que se están ejecutando. La computadora trabajará más rápidamente cuanto más memoria tenga. A menudo la cantidad de memoria es una potencia de 2, como 32 MB(megabytes) o 256MB. Algunas computadoras muy potentes (y caras) tienen varios gigabytes de memoria.

La CPU (*Central Processing Unit, Unidad Central de proceso*) es un elemento de hardware especial. Es el cerebro de la computadora, que permite ejecutar instrucciones de un conjunto de programas para hacer cálculos y procesar la información almacenada en la memoria o en las unidades de disco. Probablemente habrá oído hablar de los procesadores

INTEL, como Pentium, pero hay otros tipos de CPU que no son en absoluto como los procesadores INTEL. Por ejemplo, el PowerPC se utiliza en lugar de los procesadores Intel en muchas computadoras.

TESIS CON
FALLA DE ORIGEN

Está basado en un tipo de arquitectura de computadoras totalmente diferente, llamado RISC (*Reduced Instruction Set Computers*, computadoras de conjunto de instrucciones reducido). Las computadoras de Sun Microsystems utilizan otro tipo de procesador llamado SPARC (*Scalable Performance Architecture*, arquitectura de rendimiento graduable). Algunas computadoras tienen varias CPU para mejorar su rendimiento.

Una computadora puede contener otros componentes del sistema, como una o varias unidades de cinta magnética, unidades de disquete o unidades de CD-ROM.

Estos componentes también pueden llamarse dispositivos periféricos y dispositivos auxiliares, pero estos términos se reservan a menudo para dispositivos exteriores como los terminales, las impresoras y los modems.

1.3.- El Software en general

Un conjunto de instrucciones para la computadora se llama *programa*. Los programas residen en el disco duro, se ejecutan desde la RAM, controlan la CPU y utilizan dispositivos periféricos. Para distinguir estas piezas de programas de las piezas de hardware de la computadora, se les llama software. El software de la computadora se puede agrupar en las siguientes categorías:

- ☉Aplicaciones
- ☉Lenguajes
- ☉Sistemas Operativos
- ☉Utilerías

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN

Las aplicaciones son programas diseñados para proporcionar al usuario un conjunto determinado de posibilidades. Una aplicación de procesador de texto, por ejemplo, permite al usuario escribir, modificar e imprimir documentos. Los programas de aplicación están disponibles para casi todos los aspectos empresariales, como contabilidad, pagos, inventario, puntos de venta y agenda.

Los programas de aplicación también se pueden encontrar para cálculos científicos o financieros, para uso doméstico y personal, o para juegos. La persona que utiliza el programa de aplicación se llama *usuario final*.

Los lenguajes de programación son programas de software que permiten a unas personas, llamadas *programadores*, crear programas de computación.

Los lenguajes de programación como BASIC, FORTRAN O COBOL existen desde hace muchos años, y se siguen utilizando hoy en día. Se han desarrollado nuevos lenguajes de programación para proporcionar a los programadores mejores herramientas de programación como son los depuradores, editores de sintaxis y precompiladores.

El sistema operativo es el programa de control central para toda la computadora que controla los trabajos que se ejecutan. También organiza el disco duro en archivos, donde un archivo es una colección de bloques de disco que almacena la información que el usuario quiere mantener junta. El sistema operativo permite dar un nombre a cada archivo, llamado nombre de archivo, que facilita que el usuario le indique a la computadora a qué bloques del disco quiere acceder. Los archivos se pueden agrupar en directorios que, a su vez, pueden agruparse en directorios mayores.

Las utilerías son pequeños programas independientes que, normalmente, tienen un campo

de acción reducido y facilitan una única funcionalidad o una única área de funcionalidad; por ejemplo, la utilidad de hacer una copia de seguridad.

1.4.- Tipos de Software

En la actualidad existen dos ramas o tipos de software en donde unos u otros pueden ser clasificados. Existe el Software con CopyRight (Licencias y restricciones de uso) y el FreeSoftware (Software Libre). El Software con Copyright también se le conoce como comercial y es uno de los más difundidos, en gran parte por las labores de mercado que hacen sus productores. El FreeSoftware está causando una verdadera revolución desde hace una década, pero existe desde hace ya varias.

1.5- Inicio y Razón del FreeSoftware (Software Libre).

TESIS CON
FALLA DE ORIGEN

ORÍGENES DEL FREESOFTWARE

FSF (Free Software Foundation) la primera comunidad que comparte el software por Richard Stallman (publicado originalmente en el libro Open Source).

“ Cuando comencé a trabajar en el Laboratorio de Inteligencia Artificial del MIT en 1971, me incorporé a una comunidad que compartía el software que ya tenía varios años de existencia. El acto de compartir software no estaba limitado a nuestra comunidad en particular; es tan antiguo como las computadoras, de la misma manera que compartir recetas es tan antiguo como cocinar. Pero nosotros lo hacíamos en mayor grado que la mayoría de los otros.

El Laboratorio de IA usaba un sistema operativo denominado ITS (Incompatible Timesharing System) [Sistema incompatible de tiempo compartido] que los hackers² del staff habían diseñado y escrito en lenguaje ensamblador para la PDP-10 de Digital, una de las más grandes computadoras de la época. Mi trabajo como miembro de esta comunidad, como hacker de sistema en el staff del laboratorio de IA, era mejorar este sistema. No denominábamos «software libre» a nuestro software porque dicho término no existía; pero eso es lo que era.

Cuando alguien de otra universidad o compañía deseaba portar y usar un programa, lo permitíamos con gusto.

Si usted veía a alguien usando un programa interesante y poco conocido, siempre se podía pedir el código fuente para verlo, de manera que uno podía leerlo, cambiarlo, o canibalizar ciertas partes del mismo para hacer un nuevo programa.

El colapso de la comunidad

La situación cambió drásticamente durante la primera parte de los 80s cuando Digital discontinuó la serie PDP-10. Su arquitectura, elegante y poderosa en los 60s, no se pudo extender naturalmente a los espacios de direccionamiento más grandes que se hicieron factibles en los 80s.

TESIS CON
FALLA DE ORIGEN

² El uso de «hacker» para referirse al «quebrantador de la seguridad» es una confusión proveniente de los medios pasivos. Nosotros los hackers nos negamos a reconocer dicho significado, y continuamos utilizando la palabra para indicar a «alguien apasionado por la programación y que disfruta el ser hábil e ingenioso».

Esto significó que prácticamente todos los programas que componían a ITS se volvieron obsoletos.

La comunidad de hackers del laboratorio de IA ya se había colapsado, cierto tiempo antes. En 1981, la compañía derivada Symbolics había contratado a casi todos los hackers del laboratorio de IA, y la despoblada comunidad ya no era capaz de mantenerse a sí misma. (El libro Hackers, de Steve Levy, describe estos eventos, y muestra un claro panorama de esta comunidad en sus comienzos.)

Cuando el laboratorio de IA adquiere una nueva PDP-10 en 1982, sus administradores deciden utilizar el sistema no libre de tiempo compartido de Digital en lugar de ITS.

Las computadoras modernas de esa época, como la VAX o el 68020, tienen sus propios sistemas operativos, pero ninguno de ellos es software libre: usted debe firmar un «acuerdo de no revelar» (nondisclosure agreement) aún para obtener una copia ejecutable.

Esto quiere decir que el primer paso para poder utilizar una computadora era prometer que no ayudaría a su vecino. Se prohibía la existencia de una comunidad cooperativa.

La regla hecha por los dueños de software propietario era: «si usted comparte con su vecino, usted es un pirata. Si desea algún cambio, ruéguenos para que lo hagamos nosotros».

La idea de que el sistema social del software propietario--el sistema que dice que usted no tiene permitido compartir o cambiar el software-- es antisocial, que no es ético, que está sencillamente equivocado, puede ser una sorpresa para algunos lectores.

¿Pero qué otra cosa podríamos decir sobre un sistema que se basa en dividir el público e

impide socorrer a los usuarios? Los lectores que se sorprendan por esta idea es porque han tomado el sistema social del software propietario tal como se lo han dado, o porque lo han juzgado en función de los términos sugeridos por las empresas que hacen software propietario. Los publicadores de software han trabajado duro y parejo para convencer a las personas de que solamente hay una manera de ver este tema.

*Cuando los publicadores de software hablan de «hacer valer» sus «derechos» o de «detener la piratería», lo que *dicen* es secundario. El mensaje real de estas declaraciones está en las presunciones no declaradas que ellos dan por sentado; se supone que el público debe aceptarlas de manera acrítica. Así que examinémoslas.*

Una de las presunciones es que las compañías de software tienen un derecho natural incuestionable que las habilita para ser dueñas de un software, y por lo tanto a disponer de poder sobre todos los usuarios del mismo. (Si éste fuera un derecho natural, entonces sin importar cuánto daño le causara al público, no podríamos objetarlo.) De manera muy interesante, la Constitución de los Estados Unidos de América y la tradición legal rechazan esta visión; el copyright no es un derecho natural, sino un monopolio artificial impuesto por el gobierno que limita el natural derecho a copia de los usuarios.

Otra presunción no declarada es que la única cosa importante sobre del software es qué trabajo le permite realizar a usted--que a nosotros los usuarios de computadoras no nos debe importar qué clase de sociedad nos permiten tener.

Una tercera presunción es que no tendríamos software utilizable (o, que nunca tendríamos un programa para hacer tal o cual trabajo en particular) si no le ofrecemos a una compañía poder sobre los usuarios de dicho programa. Esta presunción puede haber sonado

plausible, antes de que el movimiento por el software libre demostrara que podemos hacer abundante software útil sin ponerle cadenas.

Si nos resistimos a aceptar dichas presunciones, y juzgamos acerca de estos temas sobre la base moral que nos da el sentido común ordinario y ponemos al usuario en primer lugar, arribaremos a conclusiones muy distintas. Los usuarios de computadoras deben tener libertad para modificar los programas para ajustarlos a sus necesidades, y libertad para compartir el software, porque la base de la sociedad está en ayudar a las otras personas.

No se dispone aquí del espacio necesario para explayarnos en el razonamiento que hay detrás de esta conclusión, y por ese motivo pido al lector que vea la página web <http://www.gnu.org/philosophy/why-free.es.html>.

Una elección moral severa.

Al desaparecer mi comunidad, se hizo imposible continuar como antes. En lugar de ello, me enfrenté a una elección moral severa. La elección fácil era unirme al mundo del software propietario, firmar los acuerdos de no revelar, y prometer que no iría en ayuda de mi amigo hacker. Es muy probable que desarrollara software que se entregaría bajo acuerdos de no revelar y de esa manera incrementar también las presiones sobre otra gente para que traicionen a sus compañeros. Podría haber hecho dinero de esta manera, y tal vez me hubiese divertido escribiendo código. Pero sabía que al final de mi carrera, al mirar atrás a los años construyendo paredes para dividir a la gente, sentiría que usé mi vida para empeorar el mundo. Ya había estado del lado en que se reciben los acuerdos de no revelar, por experiencia propia, cuando alguien se negó a entregarme, a mí y al Laboratorio de IA del MIT, el código fuente del programa de control de nuestra impresora. (La ausencia de

ciertas características en este programa hacía que el uso de la impresora fuera frustrante en extremo.) Así que no podía decirme a mí mismo que los acuerdos de no revelar son inocentes. Me enojó mucho cuando él se negó a compartir con nosotros; no podía ahora cambiarme de lugar y hacerle lo mismo a todos los demás. Otra elección, fácil pero dolorosa, era abandonar el campo de la computación. De esta manera no se usarían mis habilidades para mal, pero aún así se desperdiciarían. Yo no sería culpable por dividir y restringir a los usuarios de computadoras, pero ello sucedería igual.

Así que busqué la manera en la cual un programador podría hacer algo para bien. Me pregunté: ¿habrá algún programa o programas que yo pueda escribir, de tal manera de otra vez hacer posible una comunidad? La respuesta era clara: lo primero que se necesitaba era un sistema operativo. Este es el software crucial para empezar a usar una computadora.

Con un sistema operativo usted puede hacer muchas cosas; sin uno, ni siquiera puede funcionar la computadora. Con un sistema operativo libre, podríamos tener de nuevo una comunidad de hackers cooperando--e invitar a cualquiera a unírseos.

Y cualquiera sea capaz de utilizar una computadora sin que de movida conspire a favor de la privación de sus amigas o amigos.

TESIS CON
FALLA DE ORIGEN

Como desarrollador de sistema operativo, tengo las habilidades apropiadas para esa tarea. Así que aún cuando no tenía garantías de éxito, me di cuenta que había sido elegido para hacer ese trabajo. Decidí hacer que el sistema fuese compatible con Unix pues así sería portable, y los usuarios de Unix podrían cambiarse a él con facilidad.

El nombre GNU se eligió siguiendo una tradición hacker, como acrónimo recursivo para «GNU is Not Unix». Un sistema operativo es más que un núcleo, apenas suficiente para hacer funcionar otros programas.

En los 1970s, todo sistema operativo digno de llamarse así incluía procesadores de órdenes, ensambladores, compiladores, intérpretes, depuradores, editores de texto, programas de correo, y muchos otros. ITS los tenía, Multics los tenía, VMS los tenía, Unix los tenía. El sistema operativo GNU también los incluía.

Más adelante escuché estas palabras, atribuidas a Hillel³ :

Si yo no me preocupo por mí mismo, ¿quién lo hará por mí?

Si sólo me preocupo por mí mismo, ¿qué es lo que soy?

Si no lo hago ahora, ¿cuándo?

La decisión de iniciar el proyecto GNU se basó en un espíritu similar.

Libre como en libertad

El término «free software» [N. del T.: en inglés free = libre o gratis] se malinterpreta a veces--no tiene nada que ver con el precio.

El tema es la libertad. Aquí, por lo tanto, está la definición de software libre: un programa es software libre, para usted, un usuario en particular, si:

³ Como ateo que soy, no soy seguidor de ningún líder religioso, pero algunas veces encuentro que admiro alguna cosa que dijo uno de ellos.

Usted tiene la libertad para modificar el programa o para adaptarlo a sus necesidades. (Para que esta libertad sea efectiva en la práctica, usted debe tener acceso al código fuente, porque modificar un programa sin disponer del código fuente es extraordinariamente difícil.)

Usted tiene la libertad para redistribuir copias, tanto gratis como por un canon.

Usted tiene la libertad para distribuir versiones modificadas del programa, de tal manera que la comunidad pueda beneficiarse con sus mejoras.

Como «free» [libre] se refiere a libertad y no a precio, no existe contradicción entre la venta de copias y el software libre. De hecho, la libertad para vender copias es crucial: las colecciones de software libre que se venden en CD-ROM son importantes para la comunidad, y la venta de las mismas es una manera importante de obtener fondos para el desarrollo de software libre. Por lo tanto, si la gente no puede incluir un programa en dichas colecciones, dicho programa no es software libre.

A causa de la ambigüedad de «free», la gente ha estado buscando alternativas, pero nadie ha encontrado una alternativa apropiada. El idioma inglés tiene más palabras y matices que ningún otro, pero carece de una palabra simple, no ambigua que signifique «libre», como en libertad--«unfettered» [sin cadenas] es la palabra que más se acerca en significado. Otras alternativas como liberated [liberado], freedom [libertad] y open [abierto] tienen el significado equivocado o alguna otra desventaja.

TESIS CON
FALLA DE ORIGEN

El inicio del proyecto

En enero de 1984 renuncié a mi trabajo en el MIT y comencé a escribir software GNU. Era necesario abandonar el MIT, para que el MIT no interfiriera con la distribución de GNU como software libre. Si hubiese continuado como parte del staff, el MIT podría haber reclamado propiedad sobre el trabajo, y podría haber impuesto sus propios términos de distribución, o incluso podría haberlo transformado en un paquete de software propietario. Yo no tenía la intención de hacer un trabajo enorme sólo para ver que perdía la utilidad para la cual se había realizado: crear una nueva comunidad para compartir software. Sin embargo, el Profesor Winston, por entonces a cargo del Laboratorio de IA del MIT, me invitó amablemente a que continué utilizando las instalaciones del Laboratorio.

¿Es libre el programa para cualquier usuario?

Si un programa es software libre cuando abandona las manos de su autor, esto no significa que será software libre para todos los que tienen una copia de él. Por ejemplo, el software de dominio público (software que no está sujeto al copyright de nadie) es software libre; pero cualquiera puede hacer una versión modificada propietaria a partir de él. En ese mismo sentido, muchos programas libres están sujetos a copyright pero se distribuyen mediante sencillas licencias permisivas que admiten las versiones modificadas propietarias. El ejemplo paradigmático de este problema es el X Window System. Desarrollado en el MIT, y entregado como software libre con un licencia permisiva, fue rápidamente adoptado por varias compañías de computación. Éstas agregaron X a sus sistemas Unix propietarios, sólo en formato binario, y lo cubrieron con el mismo acuerdo de no revelar.

Estas copias de X eran tanto (software) libres en cuanto lo era el Unix.

Los desarrolladores del X Window System no consideraban que esto fuese un problema-- esperaban y buscaban que esto sucediese. Su meta no era la libertad, sólo el «éxito», definido como «tener muchos usuarios». No les preocupaba si esos usuarios tenían libertad, sólo que sean numerosos.

Esto nos lleva a una situación paradójica en la cual dos maneras distintas de contabilizar la cantidad de libertad dan por resultado dos respuestas distintas a la pregunta «¿Es libre este programa?». Si usted juzga en base a la libertad que se proporcionaba con los términos de distribución de la entrega del MIT, diría que X es software libre. Pero si usted mide la libertad del usuario promedio de X, diría que X es software propietario. La mayoría de los usuarios de X usan las versiones propietarias que vienen con los sistemas Unix, no la versión libre.

TESIS CON
FALLA DE ORIGEN

Copyleft y la GNU GPL

La meta de GNU era dar libertad a los usuarios, no sólo ser popular. Por lo tanto, debíamos usar términos de distribución que impidieran que el software GNU se transformara en software propietario. El método que utilizamos se denomina «copyleft»¹.

El copyleft usa la ley de copyright, pero le da vuelta para servir a lo opuesto de su propósito usual: en lugar de ser un medio de privatizar el software, se transforma en un medio de mantener libre al software.

¹ En 1984 o 1985, Don Hopkins (un compañero muy imaginativo) me envió una carta por correo. En el sobre, escribió varios dichos divertidos, entre ellos éste: «Copyleft—all rights reversed» [Copyleft--todos los derechos reversados]. Ubicé la palabra «copyleft» para denominar al concepto de distribución que estaba desarrollando en esa época.

La idea central del copyleft es que le damos a cualquiera el permiso para correr el programa, copiar el programa, modificar el programa y redistribuir versiones modificadas--pero no le damos permiso para agregar restricciones propias. De esta manera, las libertades cruciales que definen al «software libre» quedan garantizadas para cualquiera que tenga una copia; se transforman en derechos inalienables. Para que el copyleft sea efectivo, las versiones modificadas deben ser también libres. Esto asegura que todo trabajo basado en el nuestro quedará disponible para nuestra comunidad si se publica.

Cuando los programadores que tienen trabajo como programadores se ofrecen como voluntarios para mejorar un software GNU, es el copyleft lo que impide que sus empleadores digan: «no puede compartir esos cambios, porque los queremos usar para hacer nuestra versión propietaria del programa».

El requerimiento de que los cambios deben ser libres es esencial si queremos asegurar la libertad para cada usuario del programa. Las compañías que privatizaron el X Window System en general realizaron algunos cambios para transportarlo a sus sistemas y hardware.

TESIS CON
FALLA DE ORIGEN

Estos cambios fueron pequeños comparados con el gran tamaño de X, pero no fueron triviales. Si el hacer cambios fuera una excusa para negar libertad a los usuarios, sería fácil para cualquiera tomar ventaja de la excusa. Un tema relacionado trata la combinación de un programa libre con código no libre. Tal combinación será inevitablemente no-libre; cualesquiera libertades que falten a la parte no-libre, le faltarán también al todo.

Si se permiten tales combinaciones se abriría un agujero lo suficientemente grande como para hundir el barco. Por ello, un requerimiento crucial para el copyleft es que se tape este hoyo: cualquier cosa agregada a o combinada con un programa bajo copyleft debe ser tal que la versión combinada total sea también libre y bajo copyleft. La implementación específica de copyleft que usamos para la mayoría del software GNU es la Licencia Pública General de GNU (GNU General Public License) o LPG GNU para abreviar. Tenemos otras clases de copyleft que se usan en circunstancias específicas. Los manuales GNU también están bajo copyleft, pero utilizamos un copyleft mucho más simple, porque no es necesaria la complejidad de la LPG GNU para los manuales.

La Fundación para el Software Libre

A medida que el interés en el uso de Emacs crecía, otras personas se involucraron en el proyecto GNU, y decidimos que era el momento de buscar fondos nuevamente. Por ello en 1985 creamos la «Free Software Foundation» [Fundación para el Software Libre--FSL], una organización de caridad libre de impuestos para el desarrollo del software libre. La FSL también acaparó el negocio de distribución en cinta de Emacs; más adelante lo extendió al agregar otros productos de software libre (tanto GNU como no-GNU) a la cinta, y con la venta de manuales libres.

La FSL acepta donaciones, pero la mayoría de sus ingresos han provenido siempre de las ventas--de copias de software libre, y otros servicios relacionados.

TESIS CON
FALLA DE ORIGEN

En la actualidad vende CD-ROMs de código fuente, CD-ROMs con binarios, manuales agradablemente impresos (todos con libertad para redistribuir y modificar), y las Distribuciones De Lujo (en las cuales incorporamos toda la colección de software lista para usar en la plataforma de su elección).

Los empleados de la Fundación para el Software Libre han escrito y mantenido una cantidad de paquetes de software GNU. Dos notables casos son la biblioteca C y el shell. La biblioteca C de GNU es lo que usa todo programa que corre en un sistema GNU/Linux para comunicarse con Linux. Fue desarrollada por un miembro del staff de la Fundación para el Software Libre, Roland McGrath. El shell que se usa en la mayoría de los sistemas GNU/Linux es BASH, el Bourne Again SHell³, que fue desarrollado por Brian Fox, empleado de la FSL.

Hemos provisto los fondos para el desarrollo de esos programas porque el proyecto GNU no se queda solamente en herramientas o un entorno de desarrollo. Nuestra meta era tener un sistema operativo completo, y esos programas eran necesarios para esa meta

Asistencia para el Software Libre

La filosofía del software libre rechaza una práctica específica de negocio ampliamente difundida, pero no está contra el negocio. Cuando los negocios respetan la libertad de los usuarios, les deseamos éxito.

TESIS CON
FALLA DE ORIGEN

³ «Bourne again shell» es una broma sobre el nombre «Bourne Shell», que era el shell usual en Unix.

La venta de copias de Emacs demostró una clase de negocio con software libre. Cuando la FSL se apropió de ese negocio, necesité de otro medio de vida. Lo encontré en la venta de servicios relacionados con el software libre que había desarrollado.

Esto incluía la enseñanza, sobre temas tales como cómo programar GNU Emacs, y cómo personalizar GCC, y desarrollo de software, en la mayor parte transportar GCC a otras plataformas.

En la actualidad cada una de esas clases de negocios con software libre está puesta en práctica por una cantidad de corporaciones. Algunas distribuyen colecciones de software libre en CD-ROM; otras venden asistencia en niveles que van desde responder preguntas de usuarios, reparación de errores, hasta el agregado de nuevas características mayores. Incluso estamos viendo compañías de software libre basadas en el lanzamiento de nuevos productos de software libre. Aunque, tenga cuidado—una cantidad de compañías que se asocian a sí mismas con el término «open source» en realidad basan su negocio en software no-libre que trabaja con software libre. Ellas no son compañías de software libre, sino compañías de software propietario cuyos productos tientan a los usuarios a abandonar su libertad. Ellas usan la denominación «valor agregado» lo que refleja los valores que desearían que adoptemos: conveniencia por encima de libertad. Si valoramos más la libertad, deberíamos denominarlos productos con «libertades sustraídas».

Debemos hablar acerca de la libertad

En la actualidad se estima que hay unos diez millones de usuarios de sistemas GNU/Linux, tales como el Debian GNU/Linux y Red Hat Linux. Las buenas consecuencias de esto son evidentes: mayor interés en el desarrollo de software libre, más clientes para empresas de software libre, y mayor capacidad para animar a las compañías a que desarrollen productos de software libre, en lugar de productos de software propietario. Pero el interés en el software crece más rápido que la conciencia acerca de la filosofía sobre la cual está basado, y esto crea problemas. Nuestra capacidad de enfrentar los desafíos y amenazas que se describieron más arriba depende de la voluntad de mantenerse firmes del lado de la libertad. Para asegurarnos de que nuestra comunidad tiene esta voluntad, necesitamos esparcir la idea entre los nuevos usuarios a medida que ellos llegan a nuestra comunidad. Pero estamos fracasando en esto: los esfuerzos realizados para atraer nuevos usuarios a nuestra comunidad sobrepasan por lejos a los esfuerzos dedicados a la enseñanza cívica acerca de nuestra comunidad. Necesitamos hacer ambas cosas, y es necesario que mantengamos ambos esfuerzos balanceados. En la actualidad, con frecuencia no soy el único. Es un consuelo y un placer cuando veo un regimiento de hackers excavando para mantener la trinchera, y caigo en cuenta que esta ciudad sobrevivirá--por ahora. Pero los peligros son mayores cada año que pasa, y ahora Microsoft tiene a nuestra comunidad como un blanco explícito. No podemos dar por garantizado el futuro en libertad. ¡No lo dé por garantizado! Si usted desea mantener su libertad, debe estar preparado para defenderla." (www.gnu.org)

TESIS CON
FALLA DE ORIGEN

El software libre ha marcado una nueva generación de sistemas informáticos, así como un nuevo esquema de desarrollo de aplicaciones y sistemas, demostrando que no es necesario formar parte de una gran empresa con miles de desarrolladores y grandes sumas de dinero presupuestando el desarrollo de productos, ha dejado en claro se un esquema con muchas posibilidades de triunfo, además de respetar la libertad de uso al adquirir cualquier tipo de software por cualquier medio, garantizando la NO pérdida de su Libertad.

El proyecto GNU ha dado la vuelta al mundo sin otra cosa que la difusión por parte de los que nos sentimos parte del proyecto, y este siempre estará abierto a cualquier persona que si quiera incluir, dando soluciones reales y de muy alta calidad para problemas que tal vez necesitarían sumas grandes de dinero para poder ser resueltos, además de otorgar la confianza a los desarrolladores de todo el mundo para modificar los sistemas de acuerdo a las necesidades propias.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO II

INTERNET

El internet vino a marcar mas de una generación, es considerado como uno de las tecnologías mas importantes para el público en general, una herramienta que es utilizada tanto estudiantes de cualquier nivel utilizan para llevar a cabo sus trabajos escolares, como científicos que desean compartir sus conocimientos con otros colegas o empresas que realizan en el sus transacciones comerciales.

2.1.- TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) es el lenguaje básico de comunicación o protocolo de Internet, aunque también se puede utilizar en redes privadas (Intranets). Normalmente es un programa que viene instalado en todas las computadoras. En realidad tiene dos capas. La más alta es el Transmission Control Protocol, que es la que gestiona el ensamblaje del mensaje o archivo en paquetes pequeños. La capa baja es el IP que es el que maneja las direcciones de cada paquete que se manda. Los paquetes se reúnen sólo en el destino final del mensaje. TCP/IP utiliza el modelo de comunicación cliente / servidor en el que una computadora (cliente) pide un servicio (una página web) a otra computadora(servidor) que le proporciona ese servicio a través de la red.

Ventajas de TCP/IP

- ◉Independiente del fabricante
- ◉Multiplataforma
- ◉Soporta múltiples tecnologías
- ◉Se regala y se ha convertido en un estándar

TESIS CON
FALLA DE ORIGEN

El TCP

El protocolo TCP se encarga de garantizar que lleguen todos los paquetes sin errores y de recomponer el mensaje en la recepción. O sea, se ocupa de garantizar la fiabilidad de la transmisión. TCP fragmenta y nombra los paquetes que luego transporta el IP.

El IP

Cada computadora conectada a Internet tiene una dirección IP única, es un número de 32 bits, que se escribe en grupo de cuatro números separados por puntos y comprendidos entre el 0 y el 255 (Ej.: 162.113.4.32). La dirección IP es la que permite el direccionamiento de la información a través de Internet. El protocolo IP funciona sobre una red de conmutación de paquetes. La información se fragmenta en pequeños trozos o paquetes (unos 1500 caracteres) que se envían por separado a través de la red. Cada paquete es enviado con la dirección de la computadora donde ha de ser entregado y, de forma similar a como funciona un sistema postal, cada paquete viaja independientemente de los demás por la red hasta alcanzar su destino.

2.2.- SLIP/PPP

Los usuarios que se conectan desde el hogar además lo hacen utilizando los protocolos Serial Line Internet Protocol (SLIP) o Point-to-Point Protocol (PPP). Estos protocolos encapsulan los paquetes IP para que puedan ser enviados a través de la línea telefónica normal.

2.3 DNS

El sistema de dominio de nombres es la forma en que los nombres de dominio son localizados y traducidos en una dirección IP. Es una forma de recordar fácilmente los "nombres" de las máquinas.

Las listas de dominio de nombres y sus correspondientes direcciones IP están distribuidas en una jerarquía de autoridades. Un nombre de dominio localiza una organización u otra entidad en Internet.

Ej.: www.uc3m.es Ej: rayuela.uc3m.es

Las direcciones de dominio se leen de derecha a izquierda. El dominio más importante es el último (es, com, etc.) El dominio de segundo nivel, junto con el de primer nivel define la organización o entidad. Ej: uc3m.es

El tercer nivel se define para identificar una máquina particular. El tercer nivel no es obligatorio y puede utilizarse con niveles de subdominio.

Ej: uc3m.es, es válido Ej: www.bib.uc3m.es, es válido.

Los nombres de dominios de segundo nivel son únicos y los asigna el ICANN o una de sus organizaciones. En la web, el nombre del dominio es la parte del URL que le dice al servidor de nombres de dominio cuando y dónde reenviar una petición para una página web. Una dirección IP puede tener múltiples dominios. Ej: www.bib.uc3m.es = rayuela.uc3m.es = biblioteconomia.uc3m.es

<i>Servicio</i>	<i>Protocolo</i>
Correo Electrónico	SMTP (Simple Mail Transfer Protocol)
Boletines Electrónicos de noticias	NNTP (Network News Transfer Protocol)
Información sobre usuarios	Whois, finger, X.500
Conversación textual interactiva	IRC (Internet Relay Chat)
Emulación terminal remoto	Telnet o rlogin
Transferencia de ficheros	FTP (File Transfer Protocol)
Búsqueda de archivos	Archie
Árbol de menús y visualización de archivos	Gopher
Navegación hipermedia	HTTP (Hypertext Transfer Protocol)
Búsqueda en bases de datos	WAIS (Wide Area Information Systems)
Impresión remota	LPR
Compartir volúmenes o directorios	NFS

2.4 World Wide Web

El World Wide Web (Web) es una red de recursos de información. Se basa en tres mecanismos que permiten que estos recursos sean accesibles para su lectura a la mayor cantidad posible de audiencia.

- 1.Un esquema uniforme de nombres para localizar los recursos en el web (Ej.: URLs)
- 2.Protocolos, para acceder a los recursos nombrados y que están en la web (Ej.: http)
- 3.Un lenguaje hipertextual, para navegar fácilmente entre los recursos (ej: html)

2.4.1 Los URI's

Todos los recursos disponibles en el Web, un documento html, una imagen, un video clip, un programa, etc, tienen una dirección que puede ser codificada mediante el Identificador de Recurso Universal, URI (Universal Resource Identifier). Un URI consta de tres partes:

- 1.El nombre del esquema del mecanismo que se utiliza para acceder al recurso. (ej: http)
- 2.El nombre de la máquina en la que está el recurso. (Ej: rayuela.uc3m.es)
- 3.El nombre y la ruta del recurso. (Ej: /cursos/clase1.html)

Sin embargo, el URI es un concepto mucho más amplio. Nos debemos familiarizar más con el concepto de URL, Localizador de Recurso Universal (Universal Resource Locator). El URL es un subconjunto del esquema de nombres de URI. Los URL se refieren normalmente a un recurso pero también se pueden referir a un fragmento, a una parte de ése recurso, en este caso el URL finaliza con el signo " #" al que le sigue un texto que representa el nombre de ese fragmento del texto. (Ej: <http://rayuela.uc3m.es/cursos/clase1.html#epígrafe-2>)

TESIS CON
FALLA DE ORIGEN

Los URL pueden ser de dos tipos: absolutos y relativos. Son absolutos cuando se respeta la sintaxis completa del nombre del recurso. Son relativos cuando no contienen el esquema de nombres habitual sino tan solo el nombre del recurso (ej:clase2.html). En este caso se entiende que el recurso tiene el mismo esquema y está en la misma máquina que el documento que se está visualizando en ese momento. (Ej: ../imágenes/hipertext.jpg).

En HTML los URL se utilizan para:

- ◉ Enlazar a otro documento o recurso.
- ◉ Enlazar a un hoja de estilo externa o a un script.
- ◉ Para incluir una imagen, un objeto o applets en una página.
- ◉ Para crear un mapa de imagen sensible.
- ◉ Enviar un formulario.
- ◉ Para crear un marco en un documento.
- ◉ Para citar una referencia externa.
- ◉ Para referirse a convenciones de metadatos que describen un documento.
- ◉

TESIS CON
FALLA DE ORIGEN

2.4.2 LOS PROTOCOLOS

Reglas que siguen dos computadoras para comunicarse entre sí. Ejemplos:

- ◉ <http://porky.uc3m.es/tony/tony.html>
- ◉ <https://porky.uc3m.es/tony/tony.html>

Ⓞftp://porky.uc3m.es/pub/bdocu/ala.txt

Ⓞmailto:tony@porky.uc3m.es

Ⓞtelnet://sauron.uc3m.es

Variaciones:

Ⓞnews:comp.infosystems.www.misc (para grupos de News)

Ⓞhttp://porky.uc3m.es/~tony/tony.html

Ⓞhttp://porky.uc3m.es/~tony/tony.html#prensa

Ⓞhttp://porky.uc3m.es/~tony/tony.html:8001

Ⓞfile:///c:/dohtml/prueba.htm

TESIS CON
TALLA DE ORIGEN

2.4.3 HTML

HyperText Markup Language. Lenguaje hipertexto de marcas que sirve para publicar información que pueda ser distribuida de forma global. Una especie de "lengua madre" que todos los ordenadores pueden potencialmente entender y que es el que se utiliza en el Web.

El HTML fue desarrollado por Tim Berners-Lee en el CERN y se popularizó mediante el programa Mosaic, desarrollado por el NCSA en 1990. HTML (2.0) fue aprobada en Nov. 95. HTML (3.2) fue aprobada en Enero de 1997. HTML 4.0 en abril de 1998.

2.4.4 Anclas

Un enlace es una conexión de un recurso web a otro, este concepto tan simple ha sido uno de los factores claves del éxito del Web.

Un enlace tiene dos puntos, llamados "anclas", y una dirección. El enlace comienza en el "ancla fuente" y apunta al "ancla destino", que puede ser otro recurso web o una zona o fragmento del mismo nodo o página.

La conducta asociada por defecto a un enlace es la recuperación de otro recurso web. Para activar un enlace sólo hay que pulsar sobre él con el ratón (un click).

El elemento HTML que permite incluir anclas es el elemento `<A`, uno de cuyos atributos más importantes es el `"href"` que se encarga de especificar la dirección (URL) del ancla destino.

Aunque el uso más común de un enlace es la recuperación de otro recurso Web también es posible expresar otras relaciones entre recursos.

El internet es una herramienta que obliga a que se tome en cuenta por parte de empresas e instituciones que deseen participar dentro del vertiginoso mundo de los negocios y la tecnología. Ya no es un área que solo incluya a expertos científicos o a un número reducido

de estudiantes, es ahora una herramienta que es utilizada por casi todos los ramos o categorías de público, desde amas de casa hasta empresarios buscando ofertas de productos. Por todo esto cualquier empresa o institución sería debe de tomar en cuenta el internet y hacerse presente en la red de redes.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO III

Páginas WEB

Una de las cosas que ha dado mayor empuje al internet son los documentos o páginas WEB, las cuales son tal vez el recurso mas utilizado por los usuarios al utilizar el internet. Estos documentos van desde sencillas presentaciones de información personal, hasta sitios profesionales en donde se utilizan infinidad de recursos informáticos como bases de datos, protocolos de comunicación segura o transacciones comerciales importantes para muchas empresas y compañías.

3.1.-COMO FUNCIONA LA WEB

TESIS CON
FALLA DE ORIGEN

La Red Mundial es, de lejos, la parte más popular de la Internet. Una vez que pasemos mas tiempo en la Web, la parte gráfica de la Internet, empezaremos a sentir que podemos hacer cosas sin límites. La Web permite una comunicación rica y diversa al presentar texto, gráficos, animación, fotos, sonido y vídeo.

¿Entonces en qué consiste esta milagrosa creación? Físicamente la Web está compuesta por su computadora personal, una aplicación de **navegador de Web** una conexión con un **proveedor de servicios de Internet**, computadoras llamadas **servidores** que albergan información digital, y enrutadores y conmutadores cuya función es dirigir el flujo de información.

La Web es conocida como un sistema cliente-servidor. Nuestra computadora es el cliente y la computadora remota que alberga los archivos electrónicos es el servidor. Funciona así: Digamos que queremos hacer una visita al sitio Web de Gobernación. Primero escribe la dirección o **URL** del sitio en su navegador Web (todo esto en segundos). Entonces éste solicita la página Web del servidor ubicado en México. El servidor de Gobernación envía de vuelta los datos a su computadora a través de la Internet. El navegador Web interpreta los datos y los presenta en la pantalla de nuestra computadora.

El adhesivo que mantiene ínter enlazada la Web es llamado **hipertexto e hiperenlaces**. Esta característica permite que los archivos electrónicos de la Web sean interconectados de tal manera que podamos fácilmente saltar entre ellos. En la Web, navegamos a través de las páginas de información conforme a lo que nos interesa en un determinado momento. Esto es comúnmente conocido como **explorar o navegar** la Web.

Para entrar a la Web necesitamos de un software, como el Netscape Navigator o el Microsoft Internet Explorer, conocidos como **navegadores Web**. ¿Cómo hace su navegador para distinguir entre páginas Web y otros archivos en la Internet?

Las páginas Web están escritas en un lenguaje de computadoras llamado **HTML**, que significa **Lenguaje de Marcación de Hipertexto**.

3.2.-Antecedentes de la Web.

La Red Mundial (WWW) fue originalmente creada en 1990 en CERN (en inglés), el Laboratorio Europeo de Física de Partículas. Ahora es administrado por The World Wide Web Consortium (Consortio de la Red Mundial) (en inglés), también conocido como la **Iniciativa de la Red Mundial**. El **Consortio de la WWW** es financiado por un gran número de miembros corporativos, incluyendo AT&T, Adobe Systems, Inc., Microsoft Corporation y Sun Microsystems, Inc.

Su propósito es promover el crecimiento de la Web desarrollando especificaciones y software de referencia que estarán disponibles para todos sin costo alguno. El Consortio es administrado por MIT (en inglés) con INRIA (el Instituto Nacional Francés para la Investigación en las Ciencias de la Computación) (en francés), fungiendo como un anfitrión europeo, en colaboración con CERN.

El National Center for Supercomputing Applications (Centro Nacional para las Aplicaciones de Supercómputo o NCSA) (en inglés) en la Universidad de Illinois, en Urbana-Champaign, fue instrumental para el desarrollo de las aplicaciones iniciales de software gráfico que hacían uso de las características de la Red Mundial creadas por CERN. La **NCSA** está orientada a mejorar la productividad de los investigadores al proporcionar software para fabricar prototipos científicos, análisis y visualización. La Red Mundial fue obviamente una manera de satisfacer esa misión. NCSA Mosaic (en inglés), uno de los primeros navegadores Web, fue distribuido gratuitamente al público.

Esto contribuyó directamente al fenómeno de crecimiento de la Red Mundial.

3.3.- Estructura de una página WEB.

Una **página Web** es un documento electrónico escrito en un lenguaje de computadora llamado **HTML**, o **Hypertext Markup Language** (lenguaje de marcación de hipertexto). Cada página Web tiene una dirección única, llamada **URL**, o (localizador uniforme de recursos), que identifica su localización en la Red.

Un **website** (sitio Web) tiene una o más páginas relacionadas con él, dependiendo de cómo esté diseñado. Las páginas de un sitio Web están enlazadas entre sí a través de un sistema de **hyperlinks**, (hiperenlaces), para que podamos movernos a través de ellos al hacer clic sobre un enlace. En Internet, se puede navegar a través de las páginas con información conforme a los intereses que se tengan en un momento dado.

La página Principal

Mientras se navega por la Red Mundial, veremos el término **home page** (página principal) muchas veces. Pensemos en una página principal como el punto inicial del sitio Web. Como la tabla de contenido de un libro o revista, la página principal usualmente ofrece una visión general de lo que se puede encontrar en un sitio Web. Un sitio Web puede constar de una sola página, muchas páginas, o unas pocas páginas extensas, dependiendo de su diseño. Si no hay mucha información, la página principal puede ser la única página. Pero normalmente se encuentran al menos unas cuantas páginas más.

Las páginas Web son muy distintas entre sí en lo que a diseño y contenido se refiere, pero la mayoría usa formato de revista. En la parte superior de la página hay un encabezado principal o un gráfico. Hay una lista de elementos, por ejemplo artículos, que muchas veces tienen una breve descripción. Los elementos de la lista normalmente tienen más detalles, pues están enlazados con otras páginas en el sitio Web o con otros sitios. A veces, estos enlaces son palabras resaltadas en el cuerpo del texto o están organizadas en una lista, como un índice. También pueden ser una combinación de ambas.

Una página Web también puede contener imágenes que están enlazadas con otros contenidos.

¿Cómo saber qué texto está enlazado o tiene más detalles? Los enlaces del texto aparecen de un color distinto al resto del texto, generalmente en azul y subrayados.

Cuando movemos el puntero del ratón sobre un enlace de texto o una gráfica enlazada, éste deja de ser una flecha y se convierte en una mano. A propósito, las palabras **resaltadas** suelen ofrecer pistas acerca del enlace que se va a seguir.

Cuando regresamos a una página con un enlace que ya ha visitado, las palabras de hipertexto están usualmente en otro color, para avisarnos que ya hemos visitado ese lugar. Claro que podemos ir allá otra vez.

Para estimular a los visitantes a que regresen a un sitio, algunos editores de sitios Web cambian las páginas con frecuencia. Esto hace que navegar por Internet sea tan emocionante.

Las páginas web son los elementos de la red mas "visitados" por todo tipo de usuarios, y ya no es necesario tener una computadora propia para "navegar" en internet, pues se deben de considerar los innumerables "cyber-cafés" que dan la oportunidad de visitar en cualquier momento la WEB. La palabra "navegar" ya es tan común como el de visitar una biblioteca para consultar alguna obra, o rentar una película para disfrutar de ella en el hogar, este concepto ya es de uso común entre casi toda la comunidad estudiantil desde la edad de 8 años, por ello se debe de prestar atención al desarrollo de esta importante vía de comunicación.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO IV

SQL & mySQL

Las bases de datos han facilitado el tratado de grandes cantidades de información haciendo más fácil la tarea de clasificarla y ordenarla, además de poder acceder a estos bancos de datos desde diferentes medios. La información es lo más importante que puede manejar una empresa, y si nos valemos de un medio electrónico para manipularla, se debe de contar con la garantía de que nuestra información estará segura de usuarios extraños y la disponibilidad de esta debe de ser inmediata. SQL es ya todo un protocolo para manejar bases de datos enormes, el cual es utilizado en casi todas las líneas de bases de datos comerciales y libres.

4.1.-¿Qué es SQL?

SQL es un método basado en un potente lenguaje, para organizar, administrar y consultar datos almacenados en una computadora. SQL es una sigla que deviene de su nombre en inglés "Structured Query Language" (Lenguaje de Consulta Estructurado). Más específicamente SQL esta definido en torno al modelo de bases de datos relacionales, basado en el álgebra relacional, esto le da a SQL las ventajas que lo imponen como el sistema de mayor aceptación.

TESIS CON
FALLA DE ORIGEN

Algunas de las ventajas son:

- Marco teórico sólido, fundamentado en el álgebra relacional
- Simplicidad de conceptos (modelo de base de datos: tablas = líneas x columnas)
- Definición de vínculos en la consulta, esto le da a SQL una gran flexibilidad
- Fácil y rápido aprendizaje
- Arquitectura cliente-servidor
- Integración con cualquier lenguaje de programación
- Estandarización

TESIS CON
FALLA DE ORIGEN

4.2.-¿Que es MySQL ?

MySQL es un Sistema de Gestión de Base de Datos. Una Base de Datos es una colección estructurada de datos. Puede ser, desde una simple lista de artículos a las inmensas cantidades de información en una red corporativa.

MySQL es un gestor de bases de datos SQL (Structured Query Language). Es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas / librerías). Podemos agregar, acceder, y procesar datos grabados en una base de datos.

Actualmente el gestor de base de datos juega un rol central en la informática, como única utilidad, o como parte de otra aplicación.

Es un Sistema de Gestión de Base de Datos Relacional. El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información debe estar contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos. Esto añade velocidad y flexibilidad.

MySQL es un software de código abierto esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Podemos descargar MySQL desde Internet y usarlo sin pagar nada, de esta manera cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades. MySQL usa el GPL (GNU Licencia Publica General) para definir que podemos y no podemos hacer con el software en diferentes situaciones. Entre otras cuestiones esta licencia aclara que no cuesta dinero a menos que lo incluyamos en un software comercial y tenemos el código fuente.

TESIS CON
FALLA DE ORIGEN

4.3.-¿Porque usar MySQL?

MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños. Además tiene un conjunto muy practico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones.

La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos en Internet.

TESIS CON
FALLA DE ORIGEN

4.4.-Breve Historia

IBM empezó a comercializar en 1981 el SQL y desde entonces este producto ha tenido un papel importante en el desarrollo de la bases de datos relacionales. IBM propuso y fue aceptada , una versión de SQL al Instituto de Estándares Nacional Americano(ANSI) y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. En 1983 nació DB2 la más popular(por lo menos en los grandes ordenadores) de las bases de datos de este tipo hasta estos mismos momentos. En el mundo GNU, una de las bases de datos que se reseña en cualquier referencia de aplicaciones de éste tipo bajo LINUX, es MySQL aunque no está incluida en ninguna distribución ya que no tiene licencia GNU como tal, para comercializarla a ella o a cualquier software que la utilice o se sirva de ésta habrá que adquirir una licencia.

Alrededor de la década del 90, Michael Windenis (monty@analytikema.se) comenzó a usar MySQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Sin embargo, después de algunos testeos llego a la conclusión que mSQL no era lo suficientemente rápido ni flexible para sus necesidades. De todo esto surgió en una nueva interfaz SQL (claro que con código mas portable) con algunas apariencias similares en la API de C y en los nombres y funciones de muchos de sus programas. Esto había sido hecho para lograr

con relativa facilidad portar aplicaciones y utilidades de MiniSQL a MySQL.

El Origen del nombre MySQL no esta perfectamente claro. Algunos lo atribuyen al hecho de que un gran numero de nuestras librerías y herramientas le asignamos el prefijo "My" por costumbre. Sin embargo la hija de 'Monty' es además llamada My. Así que cual de las dos razones da el nombre a MySQL es aún un misterio.

TESIS CON
FALLA DE ORIGEN

4.5.Principales Características

- El principal objetivo de MySQL es velocidad y robustez.
- Escrito en C y C++, testado con GCC 2.7.2.1. Usa GNU autoconf para portabilidad.
- Clientes C, C++, JAVA, Perl, TCL.
- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y S.O. distintos.
- Sistema de contraseñas y privilegios muy flexible y segura.
- Todas la palabras de paso viajan encriptadas en la red.
- Registros de longitud fija y variable.
- 16 índices por tabla, cada índice puede estar compuesto de 1 a 15 columnas o partes de ellas con una longitud máxima de 127 bytes.
- Todas las columnas pueden tener valores por defecto.
- Utilidad (Isamchk) para chequear, optimizar y reparar tablas.
- Todos los datos están grabados en formato ISO8859_1.

- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintas lenguas.
- Todos los comandos tienen -help o -? Para las ayudas.
- Diversos tipos de columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc.
- ODBC para Windows 95 (con fuentes), se puede utilizar ACCESS para conectar con el servidor.

4.6.-Sistemas operativos soportados por MySQL:

Los siguientes son solo algunos de los Sistemas operativos que soporta MySQL:

- AIX 4.x con consulta nativa.
- BSDI 2.x con el paquete MIT-pthreads incluido.
- BSDI 3.0, 3.1 y 4.x con consulta nativa.
- DEC Unix 4.x con consulta nativa.
- FreeBSD 2.x con el paquete MIT-pthreads incluido.
- FreeBSD 3.x y 4.x con consulta nativa.
- HP-UX 10.20 con el paquete MIT-pthreads incluido.
- HP-UX 11.x con consulta nativa.
- Linux 2.0+ con LinuxThreads 0.7.1+ o glibc 2.0.7+ .
- Mac OS X Server.

- ↪ NetBSD 1.3/1.4 Intel y NetBSD 1.3 Alpha (carece de la marca GNU).
- ↪ OpenBSD > 2.5 con consulta nativa. OpenBSD < 2.5 con el paquete MIT-pthreads incluido.
- ↪ OS/2 Warp 3, FixPack 29 y OS/2 Warp 4, FixPack 4.
- ↪ SGI Irix 6.x con consulta nativa.
- ↪ Solaris 2.5 y con consulta nativa sobre SPARC y x86.
- ↪ SunOS 4.x con el paquete MIT-pthreads incluido.
- ↪ SCO OpenServer con un puerto reciente al paquete de FSU Pthreads.
- ↪ SCO UnixWare 7.0.1.
- ↪ Tru64 Unix
- ↪ Win95, Win98, NT, y Win2000.

La versión libre fue escrita por Michael Windenis, la versión comercial es distribuida por TCX Datakonsulter AB.

El gestor de base de datos MySQL nos brinda la oportunidad de contar con un sistema confiable y muy rápido para manipular, además de que esta bajo la licencia GPL, la cual nos permite utilizarla en cualquier entorno de trabajo sin ninguna restricción. Si duda es una de las mejores opciones que tenemos para manejar una base de datos importante de forma segura y eficiente.

TESIS CON
 FALLA DE ORIGEN

Este sistema operativo era el primer sistema multiusuario y permitía llevar a cabo diferentes tareas al mismo tiempo (multitarea). Con la finalidad de medir su rendimiento inventó un juego (Space Travel) que se podía ejecutar en él. Las pruebas que realizó lo hicieron concluir que MULTICS corría muy lentamente, era ineficiente y requería de mucha memoria. Con la ayuda de Dennis Ritchie reescribió el juego para una computadora Digital PDP-7 que se encontraba abandonada y de paso desarrolló el sistema operativo mismo que recibió en 1970 el nombre de UNIX. Tanto el sistema operativo como el juego fueron desarrollados en lenguaje ENSAMBLADOR.

La gerencia de ATT se interesó en el proyecto y autorizó la compra de un equipo DEC PDP-11/20 con diferente ENSAMBLADOR. Para 1972 ya había 10 diferentes computadoras con UNIX y para 1973 Thompson y Ritchie reescribieron el kernel (programa) en un lenguaje de más alto nivel al que llamaron B, la finalidad de hacer esto se fundamentaba en no tener que reescribir el kernel para cada máquina diferente que quisieran tener con UNIX (ya que el ENSAMBLADOR depende del lenguaje interno de cada máquina). Dennis Ritchie posteriormente modificó el lenguaje B quedando registrado oficialmente como lenguaje C.

El uso de C hace de Unix un sistema operativo portable a otros equipos como HP, Digital, IBM, Apple, Sun, Intel; debido al hecho de que su compilador está accesible para diferentes sistemas operativos. Por lo anterior, originalmente hubo muchas versiones y nombres para Unix, ya que cada grupo que modificaba y compilaba el código fuente tenía un sistema operativo diferente.

ANÁLISIS CON
FUENTE DE ORIGEN

5.2.- Características Generales

Es un sistema operativo de tiempo compartido, controla los recursos de una computadora y los asigna entre los usuarios. Permite a los usuarios correr sus programas. Controla los dispositivos de periféricos conectados a la máquina.

Posee las siguientes características:

- Es un sistema operativo multiusuario, con capacidad de simular multiprocesamiento y procesamiento no interactivo.
- Está escrito en un lenguaje de alto nivel : C.
- Dispone de un lenguaje de control programable llamado SHELL.
- Ofrece facilidades para la creación de programas y sistemas y el ambiente adecuado para las tareas de diseños de software.
- Emplea manejo dinámico de memoria por intercambio o paginación.
- Tiene capacidad de interconexión de procesos.
- Permite comunicación entre procesos.
- Emplea un sistema jerárquico de archivos, con facilidades de protección de archivos, cuentas y procesos.

TESIS CON
FALLA DE ORIGEN

- Tiene facilidad para redireccionamiento de Entradas/Salidas.
- Garantiza un alto grado de portabilidad.

Diferentes usuarios pueden "ver" versiones distintas del sistema operativo debido a la capacidad del shell para configurar diversos ambientes de ejecución. Por ejemplo, se puede hacer que un usuario entre directamente a su sección, ejecute un programa en particular y salga automáticamente del sistema al terminar de usarlo.

5.3.-El núcleo del Sistema Operativo

El núcleo del sistema operativo Unix (llamado Kernel) es un programa escrito casi en su totalidad en lenguaje C, con excepción de una parte del manejo de interrupciones, expresada en el lenguaje ensamblador del procesador en el que opera.

Las funciones del núcleo son permitir la existencia de un ambiente en el que sea posible atender a varios usuarios y múltiples tareas en forma concurrente, repartiendo al procesador entre todos ellos, e intentando mantener en grado óptimo la atención individual.

5.4.- LINUX

TESIS CON
FALLA DE ORIGEN

Todo gran suceso tiene una gran historia. Linux no es la excepción. Lo que en un principio no era más que un proyecto personal de un joven que se creía el mejor programador del mundo (a sus 21 años), terminó siendo uno de los mejores sistemas operativos; usado ampliamente en todo el mundo en instituciones educativas, comerciales y gubernamentales.

5.4.1.-El Inicio

Fue en Julio de 1991 cuando un estudiante de Computer Science en Finlandia, envió su primer mensaje al grupo de noticias *comp.os.minix*, respecto a un proyecto personal sobre el sistema operativo Minix.

Este es el primer mensaje:

"From:torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroup: comp.os.minix

Subject: GCC-1.40 and a posix question

Message-ID: 1991Jul13, 100050.9886@klaava.Helsinki.FI

Date: 3 Jul 91 10:00:50 GMT

TESIS CON
FALLA DE ORIGEN

Hello netlanders, Due a project I'm working on (in minix), I'm interested in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice. Linux Torvalds torvalds@kruuna.helsinki.fi

Al que le siguió este mensaje, que muchos consideran el verdadero inicio del Linux.

From:torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroup:comp.os.minix

Subject:What would you like to see most in minix?

Summary: small poll for my new operating system

Message-ID: 1991Aug25, 20578.9541@klaava.Helsinki.FI

Date: 25 Aug 91 20:57:08 GMT

Organization: University of Helsinki.

Hello everybody out there using minix-

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix; as my OS resembles it somewhat (same physical layout of the file-sytem due to practical reasons) among other things.

I've currently ported bash (1.08) an gcc (1.40), and things seem to work. This implies that i'll get something practical within a few months, and I'd like to know what features most people want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linux Torvalds torvalds@kruuna.helsinki.fi**

****Tomado de: <http://www.cnb.uam.es/~decla/linux.htm#linus>***

TESIS CON
FALLA DE ORIGEN

Las versiones iniciales fueron distribuidas en código fuente por el propio Linus, para que otras personas puedan entender su proyecto y sobretodo para que lo ayudaran.

Linus trabajó activamente hasta la versión 0.96, pues tras ello, se sumaron al proyecto más programadores y se formó un grupo de desarrollo amplio (Linux Developers) que continúa siendo dirigido por él; pero como el mismo lo reconoce, su labor es más la de un "router" del grupo que la del desarrollo en sí.

El sistema operativo elegido para el desarrollo e implantación de este sistema ha sido Linux, ya que además de contar con toda la historia, respaldo y características de Unix, también es el modelo mas palpable del modelo GNU, es totalmente libre y es toda una aventura trabajar en este sistema lleno de sorpresas agradables y funcionalidades que hacen que cualquier programador no voltee si quiera a ver otras opciones.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO VI

HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS

Para poder desarrollar un sistema es necesario saber qué método se va a seguir por que no se puede lanzar una persona a hacer un sistema sin hacer un análisis previo para ver que realmente sea factible llevar acabo el desarrollo de un sistema.

6.1 MÉTODOS PARA EL DESARROLLO DE SISTEMAS

Cada vez mas gente se familiariza con las computadoras y poco a poco se van especializando en áreas de la computación y van observando que los programas existentes no siempre son completamente adecuados o carecen de funciones útiles para la empresa, esto ha marcado una pauta para que se desarrollen sistemas de acuerdo a las necesidades de la empresa.

6.1.1 Método del desarrollo del Análisis Estructurado

El análisis estructurado se concentra en especificar lo que se requiere que haga el sistema o la aplicación. Permite que las personas observen los elementos lógicos, lo que hará el sistema, separado de los componentes físicos como las computadoras o los medios de almacenamiento; después de esto se puede desarrollar un diseño físico eficiente para la situación donde será utilizado. (Senn, 1999:38-40)

El análisis estructurado se conforma por 3 elementos esenciales.

Descripción gráfica: la forma tradicional de describir un sistema es con un bosquejo, donde se describen sus características, su función, su interacción, etc. Pero esta forma es muy tardada y existen mejores métodos. En este caso el análisis utiliza símbolos para crear ese bosquejo pero de manera gráfica. En el análisis estructurado se hace un diagrama lógico de flujo de datos donde muestra las fuentes y destinos de los datos, identifica y da nombre a los procesos que se llevan a cabo, identifica y da nombre a los grupos de datos a los que se tiene acceso.

Diagramas de flujo de datos: es el modelo del sistema (DFD). La descripción completa del sistema está dada por el conjunto de todos los diagramas de flujo de datos.

Para desarrollar una descripción del sistema se utiliza el proceso de up-down. Esto es, se toma el modelo original se detalla en diagramas de bajo nivel que muestren características adicionales al sistema. Cada proceso se puede detallar aún mas con diagramas de flujo y se va repitiendo el mismo proceso hasta obtener un diagrama de flujo muy detallado del sistema y que el analista pueda comprender en su totalidad la parte del sistema que se está analizando.

Diccionario de datos: se definen y describen detalladamente todos los elementos en el sistema como los flujos de datos, procesos y almacenes de datos. El diccionario es importante porque si se ocupara alguna definición, un dato específico o el contenido de un flujo de datos, esta información tiene que estar en el diccionario y debe ser muy comprensible.

Diseño estructurado: se enfoca en el desarrollo de especificaciones del software. La finalidad del diseño estructurado es crear programas formados por módulos independientes unos de otros desde el punto de vista funcional, además facilita el mantenimiento de los mismos cuando surja la necesidad. Esta técnica conduce a la especificación de módulos de programa que son funcionalmente independientes. La herramienta fundamental en el diseño estructurado es el diagrama estructurado, el cual describe la interacción entre módulos independientes junto con los datos que un módulo pasa a otro cuando interacciona con él, como se muestra en la sig. Imagen:

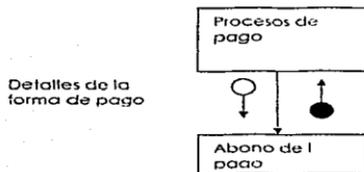


Fig. 2 Fuente: Senn. 1999:42

6.1.2 Ciclo de Vida del Desarrollo de Sistemas

Es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. En la mayor parte de las empresas las 6 actividades van muy relacionadas y es difícil establecer el orden de los pasos a seguir.

TESIS CON
FALLA DE ORIGEN

El método del ciclo de vida para el desarrollo de sistemas consta de las siguientes actividades :

1. Investigación preliminar.
2. Determinación de los requerimientos del sistema
3. Diseño del Sistema
4. Desarrollo del Software
5. Prueba de los sistemas
6. Implantación y evaluación

6.2 MODELOS PARA EL DESARROLLO DE SISTEMAS

Es importante señalar cuál es la diferencia entre un método y un modelo para el desarrollo de sistemas, ya que el analista tiene que utilizar el que mejor se adapte a sus necesidades. Por lo tanto podemos decir que el método son los pasos a seguir para realizar alguna cosa y un modelo se puede combinar con un método, mejor dicho, el modelo solo son complementos para llevar acabo una cosa mediante un método.

En este caso se utilizó un sistema que ya había sido modificado (ver capítulo VII), pero para la implantación se tomó en cuenta el modelo evolutivo.

TESIS CON
FALLA DE ORIGEN

6.2.1 Modelo Evolutivo

PROTOTIPO EVOLUTIVO

Objetivo: Poner un sistema a disposición de los usuarios finales.

Este modelo debe usarse en sistemas en los que no es posible, inicialmente, desarrollar la especificación, como por ejemplo en sistemas de inteligencia artificial.

Se basa en técnicas que permiten obtener de forma rápida las versiones del sistema.

La validación demuestra de forma subjetiva la adecuación del sistema.

Sin embargo como todo este prototipo tiene problemas, en este caso presenta los siguientes:

Los procesos de gestión existentes presuponen que el modelo de desarrollo es en cascada. Cambios continuos provocan la destrucción de la estructura del sistema, encareciendo el mantenimiento.

El tiempo de vida de los sistemas desarrollados con esta técnica, es realmente corto.

PROTOTIPO EXPERIMENTAL

Su principal objetivo es clarificar los requisitos y proporcionar a los gestores información para reducir riesgos. El prototipo se desarrolla a partir de una especificación inicial, luego se experimenta con él y finalmente se elimina.

Este prototipo no es un sistema final porque puede ignorar algunas características del sistema y como el sistema esta débilmente estructurado, es difícil de mantener.

Este modelo fue desarrollado por Boehm en 1988, con el objetivo de aunar las ventajas de los modelos de cascada y evolutivo. Hace un tratamiento explícito del riesgo del proceso. Este modelo tiene una forma de espiral, donde se puede establecer una relación entre cada giro y una fase del proceso del desarrollo.

1 ° Giro: Objetivos, alternativas, restricciones, etc. del sistema.

2° Giro: Especificación de requisitos.

3°. Giro: Diseño.

TESIS CON
FALLA DE ORIGEN

De igual manera que otros modelos, este también cuenta con diversas fases, pero no se tiene un número fijo de fases; éstas las deben decidir el equipo de gestión del proyecto:

- ✓ Definición de objetivos: identificar los objetivos de cada fase.
- ✓ Evaluación y reducción del riesgo.
- ✓ Desarrollo y validación: elegir un modelo adecuado para la siguiente fase del desarrollo.
- ✓ Planificación: se revise el estado del proyecto y se planifica el siguiente giro.

CARACTERÍSTICAS GENERALES

En cada giro se construye un nuevo modelo del sistema completo.

Este modelo puede combinarse con otros modelos de desarrollo como cascada, evolutivo, etc.

Es el mejor modelo para el desarrollo de grandes sistemas.

En este capítulo solo tratamos de justificar y adaptar uno de los modelos existentes para lograr éxito en la implantación del sistema COL. En si se utilizaron varios modelos, sin definir una estructura única, ya que este sistema pasa por 3 etapas: PHP-Nuke, COL 1.0 y la corrección COL 1.1.

Podemos concluir entonces que los sistemas no siempre siguen un solo modelo de desarrollo, sino que pueden ser una mezcla de varios, esto solo es posible si el lenguaje de programación lo permite. Infiere versatilidad y flexibilidad de las herramientas que se utilicen para desarrollar.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO VII

ANÁLISIS DEL SISTEMA COL (Control On Line)

En este capítulo trataremos de dar una idea al lector de cómo se pensó y diseñó este sistema, el cuál por estar bajo licencia GPL se presta para su implantación y post-desarrollo logrando así un desempeño eficaz. (Tomado del Proyecto de la UDV 8º semestre Informática, profesor: ing. Catalina Núñez Escamilla, Desarrollado y Documentado por: Eduardo Alonso Méndez Rodríguez, auxiliares y colaboradores: Tzararacua Bernal, Sindy Rueda, Alejandra Galván y Jesús Báez).

TESIS CON
FALLA DE ORIGEN

7.1.- Levantamiento de información del Instituto SECOMEX A.C.

El Instituto Secomex A. C. Se fundó en el año de 1996 en la Ciudad de Uruapan, Michoacán con la inquietud de dar soporte en el área de computación en asesorías, mantenimiento desarrollo de sistemas y venta de equipo. Posteriormente se crearon otros institutos en las ciudades de Morelia y Puruándiro, en dónde se consigue la incorporación a la SEP, dando así forma a lo que ahora es el Instituto.

En la actualidad el Instituto sólo funciona como tal en la ciudad de Uruapan con ubicación en la calle Obregón # 96 Altos, cuenta con un director en dónde su función principal es la coordinación de las actividades de docencia para brindar una buena capacitación, así como el cumplimiento de los objetivos de la empresa, el administrador quien controla y maneja

los recursos económicos de la Institución, el Subdirector encargado de capacitar, evaluar y coordinar al personal docente, el personal docente quienes imparten el plan de estudios a los alumnos, una secretaria quien es la encargada de archivar y cobrar las colegiaturas, su población de alumnos puede variar de acuerdo a las fechas, teniendo una capacitación semestral promedio de 250 alumnos, los cuales cuentan con un laboratorio equipado con 22 pc's, una biblioteca, 4 salones de clase, un salón para video y audio, 4 baños y una dirección, en donde su evolución ha sido en torno a los sistemas Unix y GNU software, trabajando hasta el momento con sistemas Linux.

7.2.-PLAN DE PROYECTO DE SOFTWARE

Proyecto: Sistema para Control de Calificaciones para Escuelas de Computación incorporados a estudios oficiales de la SEP.

Delimitación del medio ambiente: El Sistema será desarrollado para el Instituto Secomex, el cual se encuentra prestando sus servicios en la ciudad de Uruapan ubicado en la calle Álvaro Obregón # 96 Col. Centro.

Problemas detectados: Los maestros pasan un reporte de calificaciones a la Dirección, el cual elaboran ellos ya sea a mano o computadora, en caso de que un alumno quiera saber sus calificaciones tiene que acudir con el maestro o con la secretaria. Al consultar las calificaciones tienen que ir al archivero y buscar el expediente del alumno, por lo cual muchas veces les dicen a los alumnos que pasen hasta la siguiente semana por su calificación.

TESIS CON
FALLA DE ORIGEN

Áreas de oportunidad: Se observa que dentro del Instituto Secomex se podría mejorar el manejo de las calificaciones mediante un sistema.

Objetivos: con el sistema que proponemos se plantea:

- Eliminar el tedioso proceso de buscar calificaciones en el archivero.
- Lograr el acceso a los datos de las calificaciones de cualquier que esté conectado al servidor de la Institución
- La obtención de los datos sea fácil, rápida y confiable.

Descripción general del Proyecto: El sistema propuesto va a trabajar en un sistema Unix, conectado a un servidor de Base de Datos y podrá ser consultado desde cualquier nodo de la red, por medio de cualquier browser para web.

El sistema funciona a nivel general para cualquier institución educativa incorporada a la SEE en la especialidad de computación y aún para las no incorporadas, ya que todos llevan un mismo sistema, es decir, llevar determinada materias, maestros encargados de impartir dichas materias.

El sistema va a funcionar de manera local o remota gozando de las facilidades del sistema. Se tiene que contar con un servidor en donde debe estar instalado el sistema, el cual dará servicio de noticias, comentarios, downloads, además de representar la imagen electrónica de la institución; podrá contar además con un servicio para asignar y verificar calificaciones desde cualquier equipo que esté conectado a el servidor sin tomar en cuenta el Sistema Operativo o arquitectura que se utilice.

TESIS CON
FALLA DE ORIGEN

En dado caso de que se necesite acceder a un servicio se podrá hacer desde cualquier nodo de Internet, el servidor deberá tener una IP pública aumentando así la obtención de servicios ilimitados.

Si el equipo con el que se conecta al servidor cuenta con impresora, se podrá imprimir cualquier tipo de reporte en dicho equipo.

El sistema contará con diversos niveles de usuarios y por lo tanto diferentes niveles de seguridad restringiendo el uso de la Base de Datos.

7.3.- RECURSOS DEL PROYECTO

Requerimientos de Hardware:

ATHLON T-BIRD 1.2 GHZ

128 MB RAM

D.D. 40 GB

CD-ROM 52x

Monitor de 17"

NIC REALTEK

Tarjeta de video NVIDIA G-FORCE DDR 64 Mb

Chip de Sonido VIA

Capturadora de TV

Teclado extendido

Mouse

Lectora de 3 ½

TESIS CON
FALLA DE ORIGEN

Requerimientos de Software:

→LINUX MANDRAKE 8.1 KERNEL 2.4.1-8

→MYSQL SERVER 6.3

→PHP 4

→PHP-NUKE 5.2

→APACHE WEB SERVER 1.2.13

→BROWSER MOZILLA 1.0 o NETSCAPE 7

→KONQUEROR 2.2.1

→EMACS

→VI

→GEDIT

→BLUEFISH

Otros recursos: recursos que se necesitan para la elaboración del programa que no por que no los pusimos antes son menos importantes, ya que todos en conjunto nos servirán para llevar a cabo todo el ciclo de la elaboración del programa.

✓HUB 10Mbps

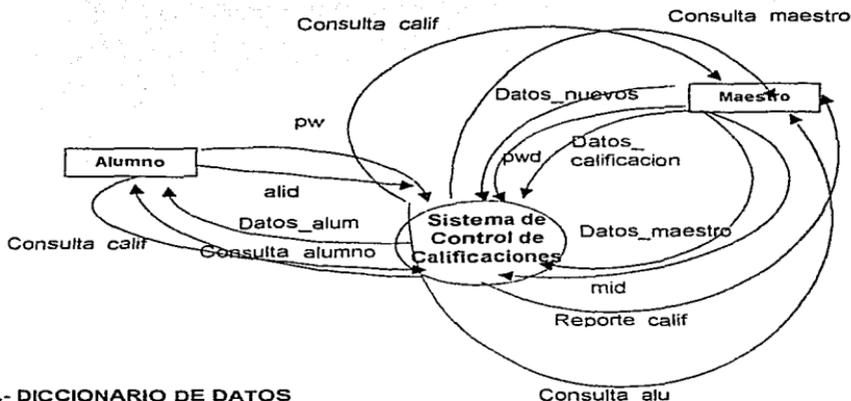
✓Impresora HP 670

✓Impresora ATI 1550

✓Documentación

TESIS CON
FALLA DE ORIGEN

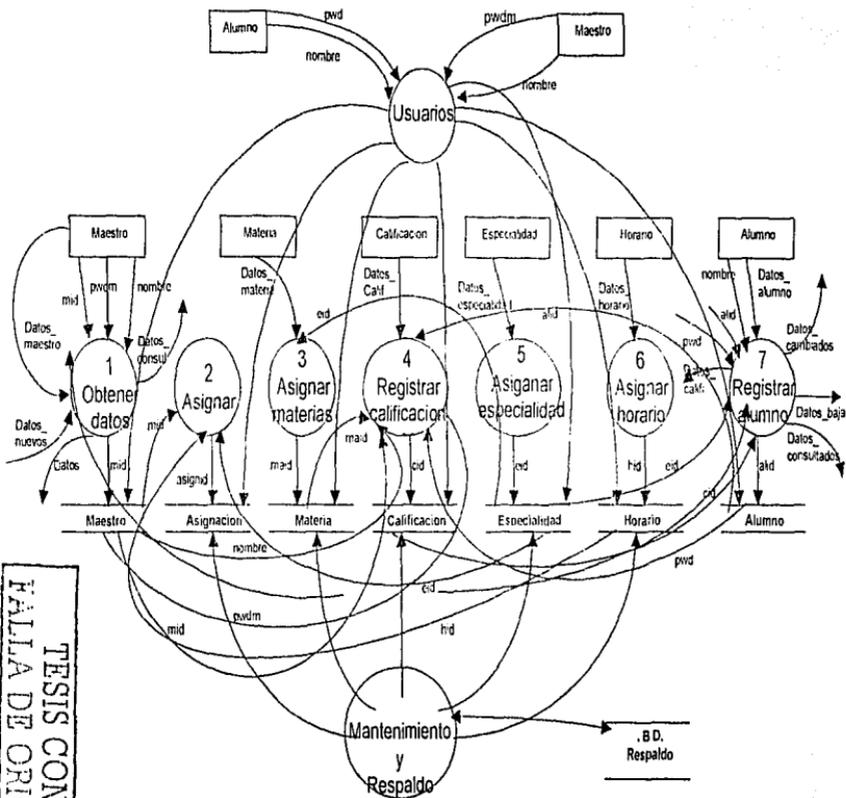
7.4.- DIAGRAMA DE CONTEXTO



7.5.- DICCIONARIO DE DATOS

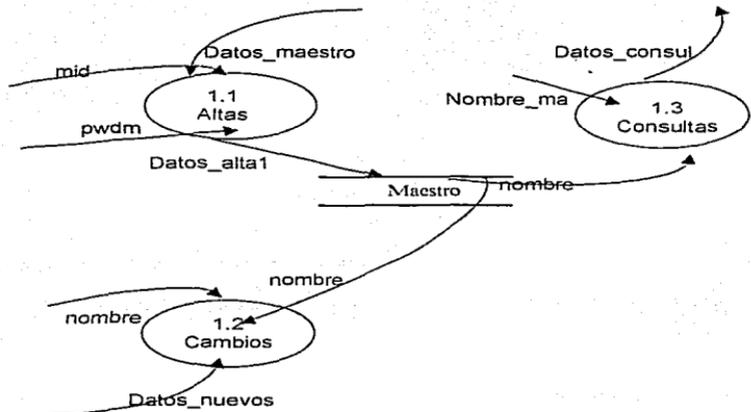
alid =	Clave primaria de la tabla alumnos
Datos_alumno =	alid+nombre+edad+direccion+telefono+pwd+uri+email+counter+hid+eid
mid =	Clave primaria de la tabla maestros
Datos_maestro =	mid+name+uri+email+pwdm+counter+radminicali
Nombre =	nombre+apellido_paterno+apellido_materno
Edad =	{entero} 2
Direccion =	Calle+numero+colonia+código_postal
Telefono =	{entero} 10
Pwd =	{caracter} 13
uri =	Dirección de alguna página de internet
Email =	{caracter} 13
Counter =	{entero} 11
Radminicali =	boolean
Consul_calif	Calificación de un solo alumno
Consulta_alumno =	Consulta de alumnos inscritos
Consulta_maestro =	Consultar información de maestros.
Reporte_calif =	Listado de las calificaciones
Datos_nuevos =	Datos_alumno

7.6.- DIAGRAMA DE NIVEL 0



**TESIS CON
FALTA DE ORIGEN**

7.7.- DIAGRAMAS DE NIVEL 1



DICCIONARIO DE DATOS

nombre =	nombre+apellido_paterno+apellido_materno
Datos =	mid+name+url+email+pwdm+counter+radmincali
mid =	Identificador de maestros
Datos_nuevos =	Datos
Datos_consul =	Datos
Datos_baja =	Datos
pwdm =	Contraseña de maestros

TESIS CON
FALLA DE ORIGEN

FORMA DE ESPECIFICACIÓN DE PROCESO

1.1

Dar de alta un Maestro

Proceso: _____

Nombre: _____

Descripción: El director teclea los datos del maestro en el formulario. Si los datos son correctos se da de alta el maestro en la base.

Flujo de Datos de Entrada	Flujo de Datos de Salida
mid Datos_maestro Pwdm	Datos_alta

Tipo de Proceso

En Línea

Por Lote

Manual

Lógica de Proceso:

HACER obtener formulario de registro de Maestro

HACER Validar si están completos los datos

SI están completos los datos ENTONCES

HACER aumentar counter

HACER almacenar en la tabla Maestro el registro correspondiente

SINO

HACER completar datos que faltan

FINSI

TESIS CON
FALLA DE ORIGEN

FORMA DE ESPECIFICACIÓN DE PROCESO

1.2

Cambiar datos de un Maestro

Proceso: _____

Nombre:

Descripción: El director modifica los datos del maestro en el formulario, en caso de sustitución de un maestro.

Flujo de Datos de Entrada	Flujo de Datos de Salida
Datos_nuevos	
Nombre	

Tipo de Proceso

En Línea

Por Lote

Manual

Lógica de Proceso:

HACER obtener formulario de cambios de Maestro

HACER Validar si están completos los datos

SI están completos los datos ENTONCES

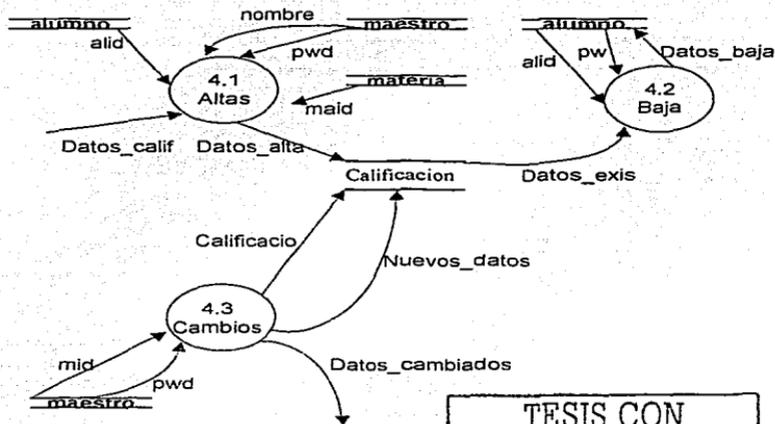
HACER cambiar los datos en la tabla Maestro en su registro correspondiente

SINO

HACER completar datos que faltan

FINSI

TESIS CON
FALLA DE ORIGEN



TESIS CON
 FALLA DE ORIGEN

FORMA DE ESPECIFICACIÓN DE PROCESO

1.3

Consultar un Maestro

Proceso: _____

Nombre: _____

Descripción: El director teclea los datos del maestro en el formulario. Si los datos son correctos se da de alta el maestro en la base.

Flujo de Datos de Entrada	Flujo de Datos de Salida
Nombre	Datos_consul

Tipo de Proceso

En Línea

Por Lote

Manual

Lógica de Proceso:

HACER introducir nombre del maestro a consultar

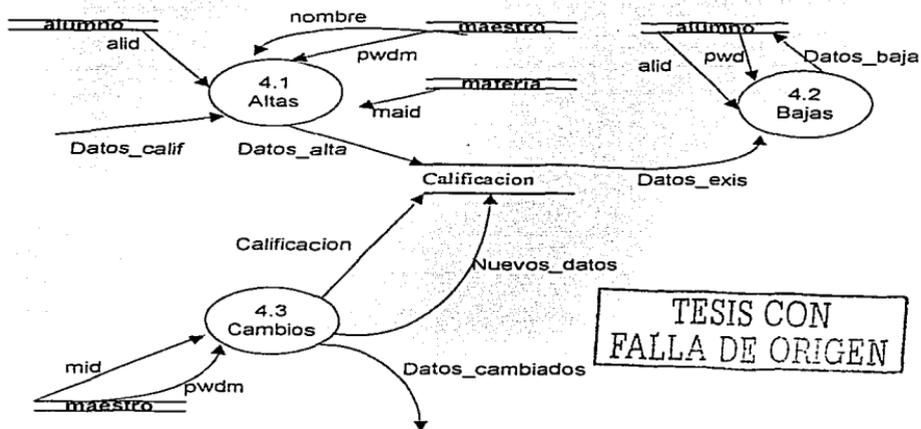
SI nombre existe en la base ENTONCES

HACER presentar la consulta del Maestro solicitado

SINO

HACER solicitar de nuevo el nombre a consultar

FIN SI



DICCIONARIO DE DATOS

Datos_alta =	cid+calificacion+cal_letra+maid+alid
Datos_exis =	Datos_alta
Datos_baja =	Datos_alta
Nuevos_datos =	cid+calificacion+cal_letra+maid+alid
calificación =	Datos_alta
Nombre =	Nompre+apellido_paterno+apellido_materno
Direccion =	Calle+numero+colonia+codigo_postal
maid =	Identificador de materia
pwd =	Contraseña de calificación
mid =	Identificador de maestros
pwdm =	Contraseña de maestro
alid =	Identificador de alumno
Datos_cambiados =	Nuevos_datos

FORMA DE ESPECIFICACIÓN DE PROCESO

4.1

Dar de alta una Calificación

Proceso: _____

Nombre: _____

Descripción: El maestro teclea las calificaciones de los alumnos correspondientes a un periodo.

TESIS CON
FALLA DE ORIGEN

Flujo de Datos de Entrada	Flujo de Datos de Salida
Datos_calif Nombre Pwdm Maid aid	Datos_alta

Tipo de Proceso

En Línea

Por Lote

Manual

Lógica de Proceso:

HACER obtener alid de la tabla alumno

HACER obtener maid de la tabla de materia

HACER introducir nombre del maestro

HACER introducir password del maestro

SI los datos son correctos ENTONCES

HACER obtener formulario de registro de calificaciones

HACER Validar si están completos los datos

SI están completos los datos ENTONCES

HACER almacenar en la tabla Calificaciones el registro correspondiente Datos_alta

SINO

HACER completar datos que faltan

FINSI

SINO

HACER corregir el nombre o password

FINSI

TESIS CON
FALLA DE ORIGEN

FORMA DE ESPECIFICACIÓN DE PROCESO

4.2

Borrar calificaciones

Proceso: _____

Nombre: _____

Descripción: El maestro puede borrar las calificaciones de los alumnos correspondientes a un periodo o de todo su historial.

Flujo de Datos de Entrada	Flujo de Datos de Salida
alid pwd Datos_baja	Datos_exis

Tipo de Proceso

En Línea

Por Lote

Manual

Lógica de Proceso:

HACER obtener alid de la tabla alumnos

HACER obtener pwd de la tabla alumnos

HACER obtener formulario de baja de calificaciones

HACER Validar si están completos los datos

Si están completos los datos ENTONCES

HACER borrar todos los datos en la tabla Calificaciones del registro especificado

SINO

HACER completar datos que faltan

FINSI

TESIS CON
FALLA DE ORIGEN

FORMA DE ESPECIFICACIÓN DE PROCESO

4.3

Cambios de las Calificaciones

Proceso: _____

Nombre: _____

Descripción: El maestro cambia las calificaciones en el formulario. Si los datos son correctos se cambian las calificaciones.

Flujo de Datos de Entrada	Flujo de Datos de Salida
mid	calificación
pwdm	Datos_cambiados
Nuevos_datos	

Tipo de Proceso

En Línea

Por Lote

Manual

lógica de Proceso:

HACER introducir nombre del maestro

HACER introducir password del maestro

HACER validar acceso

SI datos son correctos ENTONCES

HACER cambiar calificaciones del alumno

SINO

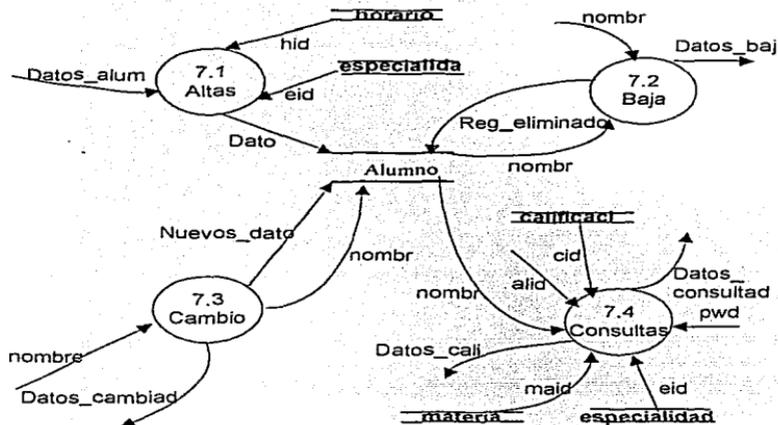
HACER solicitar otra vez el nombre

HACER solicitar otra vez el password

FINSI

TESIS CON
FALLA DE ORIGEN

TESIS CON FALLA DE ORIGEN



DICCIONARIO DE DATOS

Datos =	alid+nombre+edad+direccion+telefono+pwd+uri+email+counter+hid+eid
Nombre =	Nombre+apellido_paterno+apellido_materno
Datos_alumno =	Datos
Nuevos_datos =	alid+nombre+edad+direccion+telefono+pwd+uri+email+counter+hid+eid
hid =	Identificador de horario
eid =	Identificador de especialidad
cid =	Identificador de calificaciones

maid =	Identificador de materias
eid =	Identificador de especialidad
Datos_consultados =	Datos
Datos_califi =	Nombre+calificacion_correspondiente
Datos_cambiados =	Datos

**TESIS CON
FALLA DE ORIGEN**

FORMA DE ESPECIFICACIÓN DE PROCESO

7.1

Dar de alta un alumno

Proceso: _____

Nombre: _____

Descripción: El maestro da de alta a los alumnos con sus calificaciones correspondientes

Flujo de Datos de Entrada	Flujo de Datos de Salida
Datos_alumno hid eid	Datos

Tipo de Proceso

En Línea

Por Lote

Manual

Lógica de Proceso:

HACER obtener formulario de registro de alumnos

HACER obtener hid de la tabla horario

HACER obtener eid de la tabla especialidad

HACER escoger horario

HACER escoger especialidad

HACER validar si están completos los datos

Si están completos los datos ENTONCES

HACER almacenar en la tabla Alumno el registro correspondiente los Datos
SINO

HACER completar datos que faltan

FINSI

FORMA DE ESPECIFICACIÓN DE PROCESO

7.2

Dar de baja un alumno

Proceso: _____

Nombre: _____

Descripción: El maestro puede borrar a los alumnos que terminen su curso y aquellos que se den de baja.

Flujo de Datos de Entrada	Flujo de Datos de Salida
Nombre	Registro_eliminado Datos_baja

Tipo de Proceso

En Línea

Por Lote

Manual

TESIS CON
FALLA DE ORIGEN

Lógica de Proceso:

HACER obtener formulario de baja de calificaciones

HACER introducir nombre para dar de baja

HACER Validar si están completos los datos

SI el alumno existe en la base ENTONCES

HACER borrar todos los datos en la tabla Alumno del nombre especificado

SINO

HACER rectificar el nombre del alumno para borrar

FINSI

FORMA DE ESPECIFICACIÓN DE PROCESO

7.3

Modificar los datos del alumno

Proceso: _____

Nombre: _____

Descripción: El maestro puede cambiar los datos en el formulario. Si los datos son correctos se cambian los datos son correctos se modifica el registro del alumno especificado.

Flujo de Datos de Entrada	Flujo de Datos de Salida
Nombre Nuevos_datos	Datos_cambiados

Tipo de Proceso

En Línea

Por Lote

Manual

TESIS CON
FALLA DE ORIGEN

Lógica de Proceso:

HACER introducir nombre del alumno para modificar

SI datos son correctos ENTONCES

HACER cambiar datos del alumno

SINO

HACER solicitar otra vez el nombre del alumno para modificar

FINSI

TESIS CON
FALLA DE ORIGEN

FORMA DE ESPECIFICACIÓN DE PROCESO

7.4

Consultar los datos del alumno

Proceso: _____

Nombre: _____

Descripción: **El maestro puede consultar los datos del alumno.**

Flujo de Datos de Entrada	Flujo de Datos de Salida
cid	Nombre
eid	Datos_consultados
maid	Datos_calificacion

alid pwd	
-------------	--

Tipo de Proceso

En Línea Por Lote Manual

Lógica de Proceso:

HACER obtener cid de la tabla calificacion

HACER obtener eid de la tabla especialidad

HACER obtener maid de la tabla materia

Si los datos son correctos ENTONCES

HACER presentar datos del alumno especificado con su especialidad, materias y calificaciones al periodo correspondiente en Datos_calificacion.

SINO

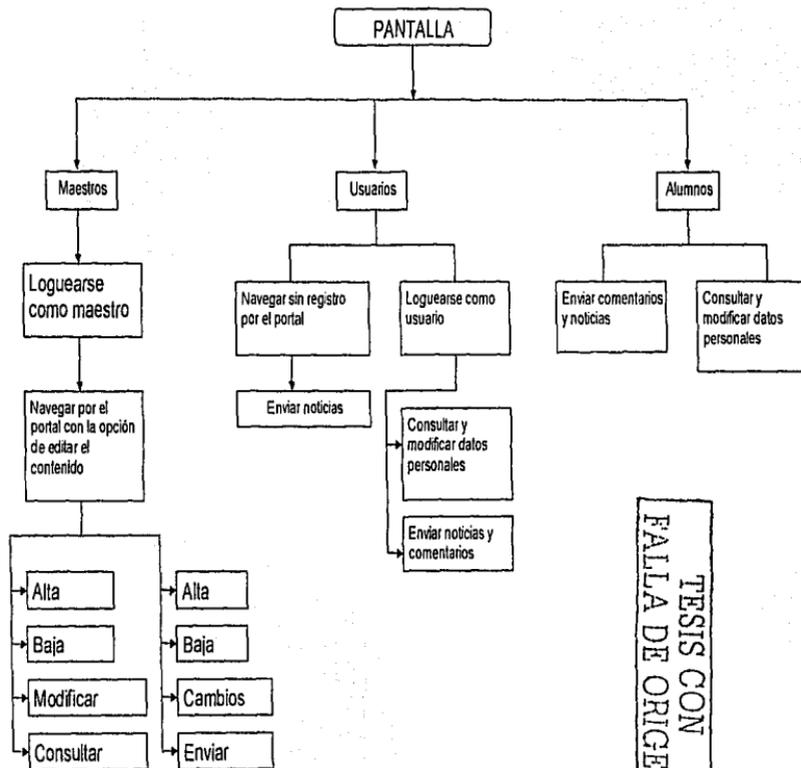
HACER rectificar los datos que sean correctos.

FINSI

TESIS CON
FALLA DE ORIGEN

ESTA TESIS NO SALE
DE LA BIBLIOTECA

7.8.- DISEÑO ARQUITECTÓNICO



TESIS CON
FALLA DE ORIGEN

FALTA
PAGINA

81

7.10 DISEÑO DE LAS PANTALLAS DE ENTRADAS Y SALIDAS

* Barnes (Opcional)

* Logo de la Empresa

* **Menú Principal**

Inicio
Opciones
Generales del
Portal

Opcionates

* **Quién está en
línea**

- Usuarios
- Conectados

* **Efemérides**

Visualización de
efemérides

* **Seleccionar
Idioma**

Lista de idiomas
disponibles

* Se utiliza en todas partes del portal sin importar el usuario que lo esté visualizando

TESIS CON
FALLA DE ORIGEN

OPCIONALES

Alumnos

Si vas a trabajar en el módulo de alumnos haz

[Click aqui](#)

Este bloque sólo se activará cuando cualquiera de los maestros se dé de alta en el sistema y sólo ellos lo podrán ver.

Noticias y Avisos

Noticias y avisos generales

Estos bloques sólo se presentarán en la página inicial del sistema sin importar usuario

Mensajes de Error

Mensajes de Errores de entrada y salida de datos

Estas ventanas se activarán sólo cuando cualquier usuario o maestro introduce algún dato que no corresponda y mandará un link para regresar a la página en la que se cometió el error.

Menús de Administración

Opciones generales de administración

Es la pantalla principal para administrar el portal.

TESIS CON
FALLA DE ORIGEN

**Datos de Alumno
para el
Administrador**

Datos para
visualizar la
información del
alumno,
calificaciones, y
modificarla o darla
de alta.

Pantalla que se activará para modificar o dar de alta los
datos e información del alumno siguiendo la estructura:

Nombre
Dirección
Teléfono
Tutor
Grupo
Horario
Profesor

Guardar

Cancelar

Calificaciones: Un link para visualizar las calificaciones

Cal 1
Cal 2
Cal 3
.
.
.
Cal N

Guarda

Cancel

TESIS CON
FALLA DE ORIGEN

Encuestas

Estos bloques serán visibles para todos los usuarios del portal y sólo se activarán en la pantalla inicial.

Información

Con que se ha creado el portal

Pantalla principal del Alumno

Datos para visualización del alumno

Ventana que se activará cuando un alumno se dé de alta en el sistema y tendrá el siguiente orden:

Nombre Dirección Teléfono Grupo
Horario Especialidad Profesor

Calif 1 Calif 2 Calif 3 Calif N

TESIS CON
FALLA DE ORIGEN

El diseño de las pantallas de alumnos va a ser desplegado junto con los datos generales de un usuario normal, ya que aunque se cuenta con dos entidades diferentes (alumnos y usuarios) contará con los mismos datos básicos agregándole al módulo de alumnos las calificaciones que le correspondan. En donde se visualizarán las siguientes datos:

1. Nombre del alumno
2. Dirección
3. Teléfono
4. E-mail (en caso de existir)
5. Tutor
6. Calificaciones

Diseño de pantalla: Cajas de texto y etiquetas con combinaciones en tonos azules y rojos.

7.11.- DIALOGOS EN LINEA

Se desplegarán los mensajes correspondientes al contenido de los campos en donde el usuario introduzca datos que no correspondan a la información solicitada como pueden ser: letras en donde no van, espacios en blanco o campos que sean requeridos y que hayan quedado vacíos. También cuando los passwords requeridos sean incorrectos, cuando los usuarios registrados, alumnos y maestros se den de baja del sistema.

TESIS CON
FALLA DE ORIGEN

7.11.1.- DISEÑO DE LA AYUDA.

La ayuda será específicamente para uso del maestro, en donde existirá un link para llevarnos a la página con los temas existentes de esta, la cual seguirá la siguiente estructura:

1.- Alumnos

Altas, Bajas. Cambios y Consultas.

2.- Calificaciones

Altas, Bajas/Cambios y Consultas.

En el caso de la ayuda para los alumnos, de igual forma que para el maestro existirá un link para llevarlos a otra página con lo temas referentes a las cosas que puede hacer el alumno, su estructura es muy simple ya que el alumno solo puede hacer consultas de sus calificaciones.

7.12.- ESTRUCTURA DE MÓDULOS Y DATOS GLOBALES

Número de módulos y finalidad:

El sistema cuenta con 20 módulos que a continuación se explicará su finalidad:

1. Maestros: administrar todas las operaciones que realizan todos los maestros.
2. Loguearse como maestro: validar el ingreso de los maestros para acceder al módulo del portal con opción de editar el contenido.

TESIS CON
FALLA DE ORIGEN

3. Editar contenido: modificar los artículos que están publicados, quitar módulos, poner quitar y modificar foros.
4. Altas alumnos: registrar a los alumnos en el sistema.
5. Baja alumnos: eliminar a los alumnos del sistema que terminan sus estudios y aquellos que se dan de baja.
6. Modificar alumnos: modificar a los alumnos que están registrados en el sistema.
7. Consultar alumnos: los maestros pueden consultar a los alumnos a los que imparte clase y que están registrados en el sistema.
8. Alta calificaciones: registrar las calificaciones de los alumnos.
9. Baja calificaciones: eliminar las calificaciones.
10. Cambios calificaciones: cambiar la calificación en caso de que haya un error.
11. Enviar noticias: una opción para que el alumno pueda expresar ideas, temas que sean de interés para el alumno.
12. Usuarios: determinar que tipo de usuario entra al sistema.
13. Navegar sin registro: cualquier usuario que no pertenezca puede ver los comentarios, las noticias y temas de interés.
14. Enviar noticias: una opción para que el usuario pueda expresar ideas, temas que sean de interés para los usuarios que visitan el portal.
15. Loguearse como usuario: el nombre de quien envía comentarios aparezca publicado, puede personalizar las opciones del portal y tener un control del usuario.
16. Consultar y modificar datos personales: consultar si sus datos personales son correctos y modificarlos si no lo son.

TESIS CON
FALLA DE CONTENIDO

17. Enviar noticias y comentarios: una opción para que el usuario pueda expresar ideas, temas que sean de interés para los usuarios que visitan el portal y su nombre se publicará
18. Alumnos: llevar el registro de los alumnos en una base de datos .
19. Enviar comentarios y noticias: una opción para que el alumno pueda expresar ideas, temas que sean de interés para el alumno.
20. Consultar y modificar datos personales: consultar si sus datos personales son correctos y modificarlos si no lo son.

7.13.-FUNCIONES DENTRO DE CADA FORMULARIO:

Alta de Alumnos.

Alta de Calificaciones.

Alta de Maestros.

Consulta/Modificación de Alumnos.

Además de visualizar los datos, se podrán utilizar la función de UpDate en la base de datos para realizar actualizaciones de datos.

Consulta/Modificación de Calificaciones.

Además de visualizar los datos, se podrán utilizar la función de UpDate en la base de datos para realizar actualizaciones de datos.

TESIS CON
FALLA DE ORIGEN

Consulta/Modificación de Maestros.

Además de visualizar los datos, se podrán utilizar la función de UpDate en la base de datos para realizar actualizaciones de datos.

Eliminar Alumnos.

Eliminar Maestros.

Eliminar Calificaciones.

7.13.1.-VARIABLES GLOBALES Y FINALIDAD

Por características del lenguaje PHP, estas no se utilizarán en nuestro sistemas ya que no representa ninguna ventaja.

En este capítulo damos un bosquejo de lo que es el diseño y funcionamiento del sistema COL, el cual se encuentra en su versión 1.0 y podemos dar paso a la implantación del sistema.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO VIII

Implantación del Sistema COL 1.0 en el Instituto SECOMEX A. C.

En este capítulo explicaremos las fases que se tienen que abarcar para llevar a cabo la implantación de un sistema de portales web, servidores y su forma de uso.

8.1 Sistema operativo e instalación de módulos necesarios.

El sistema operativo base es GNU/LINUX ya que es el único que provee de los servicios necesarios para el correcto funcionamiento de un sistema de este tipo.

SERVICIOS NECESARIOS:

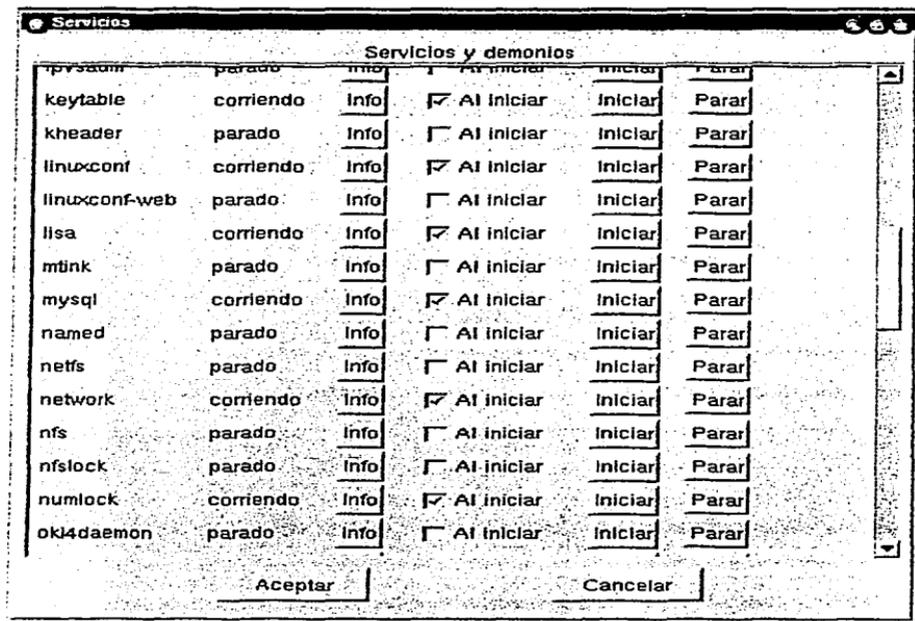
- Servidor WEB APACHE. Este módulo se recomienda instalarlo desde que se instale el sistema operativo, ya que depende de otros paquetes importantes. En la máquina en que se instaló el sistema ya contaba con este servicio, quedando como dominio: <http://scanner.isecomex.edu>. El servidor APACHE nos provee de servicios como DNS, multihosting, direccionamiento y enrutamiento ip, administración de redes y subredes, así como la asignación de dominios y subdominios en la red. Tenemos que recalcar que este servidor web debe de estar funcionando y sus servicios se deben de inicializar cuando el equipo arranque.
- Servidor Base de Datos MySQL. Este módulo es necesario para que nuestra aplicación funcione. Sin este módulo no podremos conectar nuestra aplicación hasta el servidor web. Lo más recomendable es instalar este módulo desde que se instale

TESIS CON
FALLA DE ORIGEN

GNU/Linux o FreeBSD, ya que si el módulo Mysql se instala después, este no podrá arrancar cuando el sistema operativo inicie, sino que esperará a que nosotros lo iniciemos. Para saber si el módulo está instalado y ejecutándose, podemos tratar lo siguiente:

- En la línea de comandos teclear `mysql`
- Presionar la tecla TAB
- Si se autocompleta la palabra `mysqladmin` quiere decir que el servidor esta ejecutándose y por lo tanto no tenemos que instalarlo no ejecutarlo. Ya lo hace el sistema de forma automática.
- Si no se autocompleta, entonces quiere decir que no esta ejecutándose y tal vez no este instalado, ahora vamos a comprobar que este instalado.
- Nos convertiremos en root.
- Ahora teclearemos `drakxservices`, aparecerá una lista de demonios y servicios y buscaremos `mysql`, si no está en la lista, tenemos que instalarlo, si esta en la lista, solo debemos inicializar el servicio. (ver imagen 1)
- Para realizar la instalación debemos de buscar en los cd's de nuestra distribución GNU/Linux, los paquetes `mysql.*rpm`, e instalarlos de uno en uno con la instrucción `rpm -i paquete*.rpm`, en donde `paquete*` es la cadena de caracteres que tenga el paquete como nombre. Todos los paquetes terminan con la extensión `rpm`.
- El siguiente paso será arrancar el demonio del servicio Mysql .(ver imagen 1)

TESIS CON
FALLA DE ORIGEN



(Imagen 1)

Este es un fallo del administrador de servicios, ya que nunca se logrará que arranque el servicio de base de datos automáticamente (cuestiones de seguridad).

TESIS CON
FALLA DE ORIGEN

- Módulo PHP. Indiscutiblemente este servicio se debe de activar desde la instalación del sistema operativo, ya que depende de muchas librerías, las cuales se deben de activar desde antes de la instalación. Por la experiencia que tengo, solo les podría comentar que es muy complicado activar las librerías después de la instalación y cuando lo he logrado, este módulo de repente se desactiva.

Ahora que ya tenemos todo lo necesario, pasaremos a la instalación del sistema, la cual se deben de seguir los sig pasos:

1º Adquirir derechos de superusuario (root). Solo el superusuario puede dar de alta el sistema dentro de un sistema Unix (GNU/LINUX), y ningún otro usuario podrá realizarlo.

2º Crear una carpeta en /var/www/html que servirá de hospedaje a nuestro servidor, ej:
mkdir /var/www/html/secomex.

Esta carpeta hospedará los archivos de nuestro servidor, y es muy importante contar con espacio en la partición /var, que es la que esta destinada a albergar los archivos web, ftp etc.

Para comprobar que se realizó lo correcto podemos teclear en la línea de comandos:

ls /var/www/html , al pulsar la tecla enter, debemos ver en pantalla varias carpetas, entre ellas la carpeta secomex. Es muy importante dejar este nombre de carpeta, ya que es un host dentro de nuestro servidor, ejemplo: <http://localhost/secomex/> ← este es el nombre de la carpeta, dentro del localhost que sería la carpeta /var/www/html.

TESIS CON
FALLA DE OFICEN

3º Copiar el contenido de la carpeta SECOMEX, que está en la raíz del cd-rom, hacia la carpeta que acabamos de crear. Ej: `cp /mnt/cdrom/SECOMEX/* /var/www/html/secomex`

**Recordemos que se debe de realizar una copia recursiva de los archivos, para esto debemos de teclear la instrucción: `cp /mnt/cdrom/SECOMEX/* /var/www/html/secomex -rf`*

Ahora podemos hacer un listado dentro de la carpeta verificando que todos los archivos y carpetas del sistema fueron copiados de forma correcta: `ls /var/www/html/secomex/`

4º Crear la base de datos en el servidor mysql. Ej: `mysql admin create secomex`, en donde secomex es la base de datos que damos de alta en nuestro servidor.

Para crear la base de datos, el servidor ya debe de estar inicializado, es decir, el demonio mysql debe de estar activo. Si no está activo no podremos crear la base de datos. (Ver imagen 2)

La instrucción es: `mysqladmin create secomex`, en donde secomex será el nombre de la base de datos dentro del servidor mysql. Debe de tener este nombre ya que nuestro servidor está configurado para conectarse a una base de datos con ese nombre, si no se respeta ese nombre simplemente no podrá haber ninguna conexión y el sistema no funcionará.

5º Vaciar el contenido del scrip(contiene la estructura y ordenes para crear las tablas y los campos), ej: `mysql secomex < secomex2.sql`.

TESIS CON
FALLA DE ORIGEN

Para esto debemos de entrar a la carpeta sql con la siguiente instrucción:

```
cd /var/www/html/secomex/sql
```

Ahí se encuentra el archivo `secomex.sql`, el cual es un simple scrip que contiene las instrucciones para nuestra base de datos. No se debe de utilizar otro scrip, ya que de lo contrario el sistema no podrá encontrar la estructura para la que fue diseñado.

Y listo, ahora solo tenemos que abrir nuestro navegador favorito y acceder a nuestro sistema utilizando la URL que tenga dada de alta nuestro servidor DNS, en caso de no tener ninguna, solo pondremos en la barra de navegación: `localhost/secomex/` y podremos empezar a utilizar nuestro sistema.

Para utilizar nuestro sistema debemos recordar que esta configurado como servidor, y podemos utilizarlo en la misma máquina en la que se instaló, o de forma remota en una Intranet o desde internet (solo en caso de que nuestro server tenga salida por router e ip pública).

El sistema es multiusuario, esto quiere decir que para utilizarlo debemos de introducir un nombre de usuario y una contraseña (caso administrador), si el que lo va a usar es alumno, no es necesario ninguna clave de acceso. Para entrar como maestros (administradores) y dar de alta alumnos, calificaciones etc. Debemos de hacer clic en la liga de "maestros" e introducir nuestros datos.

TESIS CON
FALLA DE ORIGEN

Inicio · Maestros · Tóp

acceso al sistema
como administrador

Admin ID freehole

Clave

Login

TESIS CON
FALLA DE ORIGEN

Ahora ya podremos controlar el portal, alumnos, calificaciones, etc.

8.2 Manejo y funcionamiento del Sistema

El portal tiene un funcionamiento muy sencillo, solo tienes que abrir un browser(navegador de internet) y escribir la url (<http://maquina.host/secomex>), o escribir la ip que la maquina tiene en la Intranet. Ej.: 192.168.0.145/secomex.

En nuestro caso el servidor esta disponible solo en la Intranet, por lo cuál cualquier máquina que este dentro de la Intranet puede acceder al servidor. El servidor requiere de

alguien que administre el portal, el cual debe de dar de alta a los profesores:

Administración de Autores

Editar Admins

God	Todo	Modificar Información	Administrador God*
freehole	Todo	Modificar Información	Borrar Autor
xpazy	Todo	Modificar Información	Borrar Autor
prueba	Todo	Modificar Información	Borrar Autor

*(La cuenta GOD no puede borrarse)

Añadir Autor

Nombre:	<input type="text"/>	(obligatorio, no podrá ser cambiado)	
Alias:	<input type="text"/>	Obligatorio	
E-Mail:	<input type="text"/>	Obligatorio	
URL:	<input type="text"/>		
Permisos:	<input type="checkbox"/> Artículos	<input type="checkbox"/> Temas	<input type="checkbox"/> Usuarios
	<input type="checkbox"/> Encuestas	<input type="checkbox"/> Secciones	<input type="checkbox"/> Enlaces
	<input type="checkbox"/> Efemérides	<input type="checkbox"/> Administrador de Archivos	<input type="checkbox"/> PUF
	<input type="checkbox"/> Descargas	<input type="checkbox"/> Reseñas	<input type="checkbox"/> Súper Usuario

En donde solo se utilizará el nombre completo del profesor, un alias(id en el sistema), e-mail, si tiene página en internet la URL, y el atributo de Súper Usuario.

También se tiene que indicar un password, el cual la primera vez será igual al alias, pero el profesor tiene que acceder al sistema cuanto antes y cambiarlo por uno que el crea conveniente. Este punto es muy importante ya que se debe de tomar en cuenta que si alguien que no sea el profesor entra con su clave y alias, podrá modificar a su antojo la información de sus alumnos.

TESIS CON
FALLA DE ORIGEN

Se debe de crear Asignaciones, las cuales hacen referencia a los horarios-especialidades que un profesor imparte. Sin las asignaciones no se podrán dar de alta ningún alumno.

Asignaciones

Nueva_Asignación

Agregar

Especialidad ▾ Horario ▾ Maestros ▾

ASIGNACIONES

EDITAR ASIGNACION

17:30 a 19:00	Sistemas Computacionales	xpazy	Modificar	Borrar
SABATINO	Sistemas Computacionales	xpazy	Modificar	Borrar
DOMINICAL	Analista en Sistemas y Diseño Gráfico	xpazy	Modificar	Borrar
16:00 a 17:30	Operador de Computadoras	xpazy	Modificar	Borrar
10:30 a 12:00	Operador de Computadoras	prueba	Modificar	Borrar

ASIGNACIONES

Para acceder a este menú solo tenemos que buscar la liga en el menú de administración con el nombre "Asignación", y nos llevará a la ventana que vemos ahora.

Ahí cualquier profesor puede asignar horarios, pero es recomendable que solo el administrador del sistema lo haga. Las asignaciones cuentan con un horario, especialidad y un profesor. Después de dar de alta la asignación, esta puede ser modificada o eliminada en el mismo menú de asignación.

TESIS CON
FALLA DE ORIGEN

Cada profesor debe de dar de alta sus alumnos, y deberá asignar las calificaciones a estos conforme las tenga listas. Ya que el profesor este con una asignación podrá añadir alumnos a la base de datos. Tendrá que hacer clic en la liga usuarios, dentro del menú de administración, y llenar un formulario con los datos de cada alumno.

Añadir Usuario

Alias	<input type="text"/>	<input type="checkbox"/> Obligatorio
Nombre	<input type="text"/>	Obligatorio
Edad	<input type="text"/>	Obligatorio
Dirección	<input type="text"/>	
Teléfono	<input type="text"/>	
Tutor	<input type="text"/>	
E-Mail	<input type="text"/>	Obligatorio
E-Mail Falso	<input type="text"/>	
URL	<input type="text"/>	
Número ICQ	<input type="text"/>	
Número AIM	<input type="text"/>	

TESIS CON
FALLA DE ORIGEN

Ya asignados los alumnos podremos dar de alta las materias, las cuales pertenecen a una especialidad u otra. En el menú de administración se encuentra la liga Materias, la cual nos llevará a la pantalla para darlas de alta:

Nueva_Materia

Especialidad	<input type="checkbox"/> --Nombre de la Materia--
Agregar	

MATERIAS

Ahora el profesor ya podrá asignar calificaciones a sus alumnos, obviamente el sistema protege al alumno y este solo puede ser visualizado por su profesor(Solo para asignar calificaciones). Pero si se quiere consultar su estadísticas cualquiera lo puede hacer.

Por último el alumno necesitará revisar sus calificaciones o imprimirlas. Para ello solo tiene que acceder al sistema de forma similar al profesor, solo que él no escribirá ningún alias ni password, y al estar en la pantalla principal hará clic en la liga de MemberList y esta lo llevará a un listado de todos los alumnos dados de alta, se hace clic en el alumno deseado y podrá ver sus calificaciones. Éste no puede realizar ningún tipo de cambio en el sistema.

En este capítulo hemos implantado nuestro sistema en un servidor, dando de alta los servicios necesarios, disponiéndonos a disfrutar del funcionamiento del sistema, dado que la forma de agregar alumnos, calificaciones, etc. Es muy sencilla, además de que la interfaz guía en todo momento al profesor, que ya debe de estar acostumbrado a este tipo de pantallas y formularios.

En el sig. Capítulo veremos cuales fallas y errores se presentaron y como podemos adecuarlo y corregirlo.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO IX

MANTENIMIENTO DEL SISTEMA COL 1.0

El desarrollo de sistemas implica una adecuada supervisión ya que cuando está funcionando en el área de batalla, es cuando se pueden detectar mas fallas, estas fallas pueden ser operacionales o propias del diseño de sistemas, y si nuestro sistema cuenta con licencia GPL, nos da la libertad de corregirlo nosotros mismos, pudiendo aprender un poco mas sobre sistemas realmente fuertes.

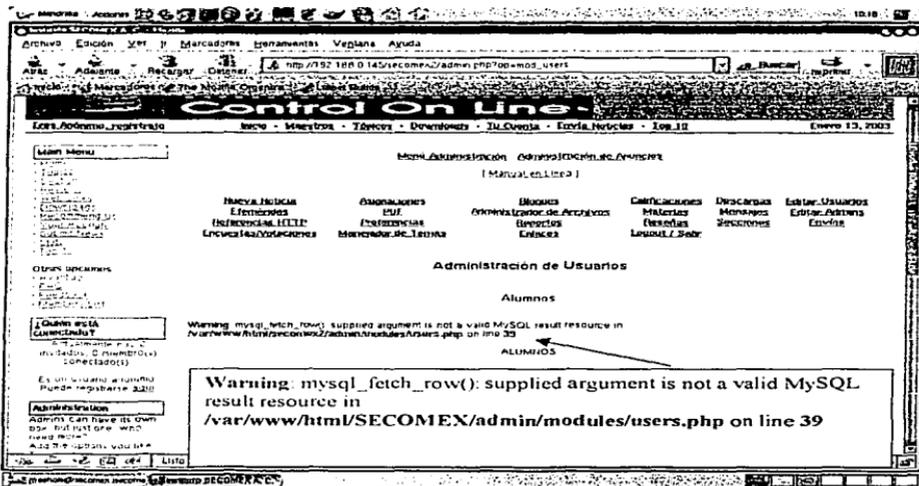
9.1 Detección de errores en COL 1.0

- a) La primer falla detectada es cuestión del diseño de la aplicación, ya que en la base de datos está definida una tabla (nuke_users) con ciertos campos que no son de utilidad, y además causan conflictos con otras tablas, los campos son hid y eid. Los cuales su función era almacenar el horario y la especialidad de cada alumno respectivamente.

Esta falla no nos permite agregar alumnos, ya que los campos antes referidos no se encuentran en la tabla, además de que si en un solo campo podemos guardar la información de dos campos, de una vez hacemos la corrección, eliminado de la base de datos los dos campos (hid, eid) y agregamos asigid, que es el que contiene los anteriores en su tabla.

TESIS CON
FALLA DE ORIGEN

En este caso podemos ver la pantalla de error así:



Se puede observar en la tabla de Alumno como trata de buscar ciertos datos, los cuales no se encuentran, y nos da una línea de código que hay que modificar... Posiblemente podríamos revisar los procesos y determinar en donde se pierden los datos, pero elegimos el camino mas corto y sencillo.

Nuestro módulo necesita de un campo, el cual relaciona a otros tres que curiosamente son los datos que necesitamos, se trata de la tabla asignación, en donde se asigna a un profesor, un horario y especialidad respectivamente.

TESIS CON
FALLA DE ORIGEN

FALTA

PAGINA

104

Ahora veamos como queda nuestra nueva base de datos:

The screenshot shows the Microsoft Access interface with the 'Mandato Linea' table structure displayed. The table has the following fields:

Nombre del campo	Tipo de datos	Clave primaria	Clave externa	Indice	Valor por defecto	Tabla vinculada	Relación
NUMERO SECCION	varchar(100)	Yes	None				
NUMERO LINEA	varchar(150)	Yes	None				
NUMERO ALUMNO	varchar(255)	Yes	None				
NUMERO MANDATO LINEA	tinyint(2)	Yes	None				
NUMERO LIBRO	int(3)	Yes	None				
NUMERO ANO	varchar(18)	Yes	None				
NUMERO SEMESTRE	varchar(25)	Yes	None				
NUMERO MATERIA	varchar(25)	Yes	None				
EDAD	varchar(40)	No	None				
ASIGNATURA	tinyint(4)	No	None		10		
ASIGNADO	varchar(10)	No	None				
UNIDAD	tinyint(1)	No	None		0		
LIBRO	tinyint(1)	No	None		0		
SEMESTRE	tinyint(1)	No	None		0		
ANIO	tinyint(1)	No	None		0		
LIBRO AÑO	tinyint(1)	No	None		0		
LIBRO SEMESTRE	tinyint(1)	No	None		0		
LIBRO ANO	varchar(255)	No	None				
LIBRO SEMESTRE ANO	int(1)	No	None		4096		
LIBRO ANO SEMESTRE	int(1)	No	None		0		
LIBRO ANO SEMESTRE ANO	varchar(30)	No	None		0		
LIBRO ANO SEMESTRE ANO SEMESTRE	tinyint(1)	No	None		0		

Los campos ya no están y podemos ver al final el campo asignid, el cual contiene una referencia a la clave primaria de asignación, para poder obtener los datos necesarios para el alumno. De esta forma solucionamos dos problemas: El diseño de nuestra base de datos es mas funcional, y las búsquedas mas rápidas, y el proceso de búsqueda no perderá datos al comunicarse en una Intranet o internet.

TESIS CON
FALLA DE ORIGEN

Esta falla de seguridad viene desde las versiones del PHP-Nuke, y son causa del usuario, ya que cuando escribimos un password el navegador nos pregunta si queremos que se nos recuerde y los usuarios sin experiencia por lo general contestan de forma afirmativa. Dado que el uso del portal para el alumno es solo de consulta de calificaciones, se desactivaron los módulos de entrada Login, para usuarios, y solo pueden hacer consultas desde el menú: Member List, en donde se encuentra la lista de todos los alumnos que están registrados.

El modelo PHP, nos presenta varias opciones para manejo de información de internet, pero a veces no son lo que una entidad puede llegar a necesitar, ahí es donde entran en juego los programadores y la licencia GPL, que nos permite manipular las líneas de código a nuestra propia conveniencia. Es total responsabilidad del programador las adecuaciones que se hagan al los sistemas y es cuando se da pie a un nuevo sistema o una nueva versión de sistema con licencia GPL.

Por los cambios realizados hasta el momento en la aplicación podemos proponer una nueva versión de este sistema: COL 1.2.

TESIS CON
FALLA DE ORIGEN

CONCLUSIONES

La empresa al contar con un servicio de estas dimensiones ha desarrollado una nueva visión a favor de la tecnología del internet, tomando en cuenta la funcionalidad de este portal, así como la nueva imagen que le da el diseño y modelo utilizado en un área en donde solo están las mejores empresas... el internet.

Los objetivos han sido cumplidos satisfactoriamente, ya que el modelo propuesto cumplió con las expectativas, logrando así la conexión hacia el Instituto desde muchas partes del mundo, fijando su imagen y cumpliendo con un excelente servicio, que de haber utilizado herramientas comerciales, en lugar del software libre, la Institución hubiera hecho un gasto muy grande, inalcanzable para la misma.

El sistema implantado aceleró y simplificó los procesos de captura de información dentro del Instituto, además de centralizar esta información en un servidor que al estar conectado a otros equipos, hace que la información este disponible en todo momento.

De igual manera este proyecto alimentó la necesidad de contar con tecnología GPL, la cuál nos permite hacer modificaciones, correcciones etc, sin entrar en problemas legales, y siempre buscando la perfección en los sistemas.

TESIS CON
FALLA DE ORIGEN

Las posibilidades aumentan para cualquier empresa al utilizar software libre, ya que los precios son mucho más bajos, además de que la calidad no se ve disminuida, sino al contrario, se puede personalizar de una forma que la calidad alcanzada a veces es mas alta que con el software propietario.

Como individuo, el software libre me da la opción de decidir libremente si lo quiero utilizar o bien escoger una opción propietaria. Si escojo la primera tendré muchas ventajas, pero si decido la segunda opción pagaré las consecuencias de una decisión que tomé libre y soberanamente, conociendo de antemano que existían otras opciones. No fui obligado a escoger la opción propietaria, ejercí mis derechos.

Como gobierno, yo tengo la obligación de escoger la opción que sea mejor para el país. Hace unos años, el software libre no era viable, pero ahora es una realidad. Como gobierno tengo la libertad de escoger entre dos opciones. Yo soy gobierno y voy a tomar las decisiones que no dañen a la sociedad civil ni a las instituciones. El software libre da libertad al gobierno para tomar decisiones conscientes y soberanas. Con el software propietario, sabemos que existen graves riesgos en la seguridad nacional, en el bienestar social y en la educación. Con el software libre, además de evitarse los riesgos anteriores, tenemos la oportunidad de trascender tecnológicamente y sobre todo, de tomar libre y soberanamente las decisiones.

TESIS CON
FALLA DE ORIGEN

En el mundo del desarrollo informático pasa algo similar. Una persona escribe un programa para resolver una necesidad específica, la comenta con otros y estos al ver que dicho programa también los beneficia, le ayudan a realizarlo. En el proceso, la propiedad intelectual no se vuelve una lucha de "egos", se transforma en un trabajo de todos y para todos; se generan líderes naturales con autoridad moral sobre el proyecto, pero nunca dueños.

Un proceso tan diferente es difícil de asimilar. Es complicado asimilar que un puñado de locos quieren regalar el resultado de su esfuerzo, sus horas de vida, su talento en algo para los demás. Imagine por un momento que todos pensáramos enloquecidamente y regaláramos parte de nuestro trabajo. Imagine que ese trabajo se transforma en algo que beneficia a su país, imagine ahora que esa locura le permite a los pequeños empresarios y emprendedores tener herramientas que no tengan restricción de uso; que tengan los códigos fuentes (la esencia del cómo fueron hechos) a la mano; que si ven algún problema, lo corrijan; que al ver que su corrección sirve, lo devuelvan a la comunidad y continúen el ciclo sin fin de esta esquizofrénica aventura...

El resultado se llama Software Libre.

TESIS CON
FALLA DE ORIGEN

Ahora vea que las personas aprenden, colaboran, no tienen miedo a ser reprendidas por usurpar programas ajenos, que no se sienten acorralados por infructuosas cacerías de brujas. Ahora vea que tiene autonomía sobre lo que quieren y hacen. Multiplique a las personas con este sentimiento y forme un territorio, ahora el cúmulo de gente siente que

su país no le debe nada a nadie, que puede crecer bajo su propio esfuerzo y a ese sentimiento llámele Soberanía Nacional...

En este trabajo damos por hecho que nuestra hipótesis ha quedado comprobada, hemos trabajado con un portal multidinámico, el cual nos brinda mucha flexibilidad en el acceso y manejo de su sistema, ya que esta tan abierto como nosotros lo queramos, además de que no tenemos la barrera de las plataformas que se utilicen, es decir, Windows®, Linux o Mac, no importa, nuestra aplicación funciona a la perfección en cualquier tipo de plataforma. H

Hemos analizado el desarrollo de una aplicación GPL, hemos visto que es muy sencillo hacerla funcionar y podemos ver también que el desempeño de los empleados y profesores del instituto en el que fue implantado es mucho mayor, haciendo que la información este donde se le necesita.

TESIS CON
FALLA DE ORIGEN

BIBLIOGRAFÍA

Moritsugu, Steve.
"Serie Práctica UNIX"
ED. Prentice Hall
2000

Senn, James A.
"ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN"
ED. Mc Graw Hill
Mayo1999

Bowen, Rich; Coar Ken.
"Servidor Apache"
ED. Prentice Hall
2000

Lehey, Greg.
"The Complete FreeBSD"
ED. Walnut Creek CDROM
1999

Conectiva.
"Gráficas en la Web con LINUX"
ED. Prentice Hall
2000

Wall, Kurt.
"Programación en Linux"
ED. Prentice Hall
2000

TESIS CON
FALLA DE ORIGEN

Sitios y páginas web de consulta:

<http://www.cnb.uam.es/~decla/linux.htm#linus>

<http://www.php.net/manual/en/ref.mysql.php>

<http://www.php.net/manual/en/ref.mcrypt.php>

<http://www.php.net/manual/en/ref.array.php>

<http://www.php.net/manual/en/ref.gettext.php>

<http://www.linux.org/docs/ldp/howto/DNS-HOWTO-3.html#ss3.1>

<http://www.linux.org/docs/ldp/howto/DNS-HOWTO.html>

<http://www.gnu.org/gnu/thegnuproject.es.html>

<http://www.gnu.org/philosophy/bsd.es.html>

<http://www.gnu.org/philosophy/microsoft.es.html>

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ipsec.html

TESIS CON
FALLA DE ORIGEN