

**UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE COMPUTACIÓN**

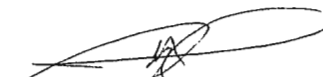
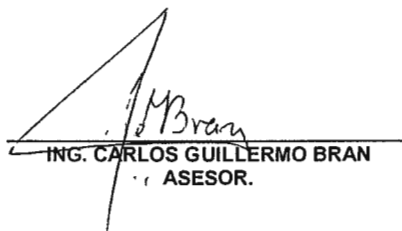
JURADO EVALUADOR

TEMA:

**ELABORACIÓN DE UNA INTERFAZ AUTOMATICA PARA LA
CREACIÓN DE CURSOS ACADÉMICOS VIA WEB.**



**ING. FRANCISCO ERNESTO CASTILLO
JURADO**


**LIC. MELVYN JOSE HERNÁNDEZ
JURADO.**
**ING. CARLOS GUILLERMO BRAN
ASESOR.**

**UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE COMPUTACION**



**"ELABORACION DE UNA INTERFAZ
AUTOMATICA PARA LA CREACION DE CURSOS
ACADEMICOS VIA WEB"**

**TRABAJO DE GRADUACIÓN PARA OPTAR AL GRADO DE
INGENIERO EN CIENCIAS DE LA COMPUTACION**



ASESOR: ING. CARLOS BRAN

INTEGRANTES

**PATRICIA JEANNETTE MOLINA MORENO
SALEM EMILIO SALVADOR SALEM ZABLAH
MARCELA IRAYDA VALIENTE PLATERO**

SOYAPANGO, ABRIL DE 2001.

Dedicatoria

A Dios Todopoderoso y a María Santísima

Por brindarme la fuerza y el entendimiento a lo largo de mi carrera y permitirme llegar hasta esta fecha, y comprender que siempre hay un momento para empezar, para levantarnos de las caídas y para emprender de nuevo el camino, por muy largo que sea.

A mi Papi Don Adalberto y a mi Mamí Doña Dorita

Porque siempre nos han enseñado a forjarnos una meta y luchar hasta conseguirla, como me lo han demostrado con su ejemplo. Por todos los sacrificios que han hecho por nosotros sus hijos, hoy les dedico este título que gracias a su amor, confianza, apoyo, comprensión, paciencia, dedicación, esfuerzos y sacrificios he logrado alcanzar. Gracias también por las alegrías y las palabras de aliento que me han brindado siempre y todos aquellos detalles que han llenado mi vida de muchos valores para alcanzar mis objetivos, que Dios les bendiga y les proteja siempre. Los Amo con todo mi corazón...

A mis Hermanos Jenniffer y Carlitos

Por todos los momentos que hemos compartido juntos, por el apoyo incondicional, los consejos, las desveladas, los "problemillas de hermanos", y todos aquellos detalles que guardo en mi corazón, los cuales son muy especiales y que se quedaran conmigo para siempre, Dios les bendiga y les proteja. Los Quiero mucho...

A Alfredo, mi novio

Por ser el amor de mi vida, mi mejor amigo, mi apoyo incondicional, con quien no necesito decir mucho para que sepa lo que me pasa, por entenderme, comprenderme, aceptarme y quererme tal como soy, gracias por ser como eres y ayudarme en todo sentido, por estar ahí cuando te he necesitado, Dios quiera que sea siempre así. Que Dios te guíe, te cuide y te bendiga siempre. Te Amo!!!!...

A mis amigos Salem y Marcela

Y además compañeros de lucha en la etapa final de nuestra carrera, gracias por los momentos que vivimos, a quienes les digo "Lo logramos, Felicidades Ingenieros!!!!..."

A mis Abuelos

Papa Meme y Mama Toya, que aunque ya no están conmigo físicamente, siempre están cuidando de mí, sé que son mis Ángeles de la guarda.

Papa Beto y Mama Tona, que Dios les cuide y les proteja siempre.

A mis amigos de siempre

Corina, Gloria Marta, Mónica, Selvint, Juan Ramón y Frank, por los momentos de estudio y los de no estudio, pero más que nada por la amistad que nos une, a quienes quiero decirles muy sinceramente que cuentan conmigo siempre...

A nuestro asesor Ing. Carlos Bran, por su paciencia y apoyo durante esta jornada.

Y a toda mi familia y demás amigos

Que no logro escribir en estas páginas, pero que los llevo en un lugar especial de mi corazón, porque sé que se alegran conmigo por este Título.

Patty Molina

DEDICATORIA

Esta tesis la dedico a Dios Todopoderoso porque me ha dado la vida, la luz y la sabiduría para finalizar esta importante etapa de mi vida. A la Virgen María, nuestro señor Jesucristo y San Juan Bosco porque siempre me acompañaron en cada paso que di.

A la memoria de mi padre, Yacooub Salem Andon, porque siempre ha sido y será el ejemplo más grande que tengo. Por todo el apoyo que me brindaste a lo largo de mi vida. Papá siempre has sido el punto de referencia que he tomado para esforzarme más cada día y dar lo mejor de mí, este triunfo es tan tuyo como mío.

A mi madre, Gloria Isabel Zablah de Salem, porque siempre estuviste a mi lado, dándome tu apoyo incondicional, tu comprensión y tu amor. Mamá siempre buscabas la forma de minimizar mi cansancio. Sin ti no habría finalizado esta etapa de mi vida.

A mis hermanas, Sonia, Yasmín y Jeannette, gracias por haberme comprendido en esos duros momentos.

A mis Abuelitos, Gloria y Emilio, inspiración de mi trabajo y máximo ejemplo de la capacidad humana, amor y comprensión.

A mi tío, Gerardo, gracias por tus consejos cuando más los necesite.

A mi Esposa, Marleny, Gracias Amor, porque en los duros momentos, siempre estuviste a mi lado, llenándome de tu amor y tu paz, sembrando la semilla de esperanza cuando más la necesite. Mi triunfo es el fruto de tu amor, tu comprensión. Eres el mejor regalo que la vida me pudo dar. Te Amo.

A mi Hijo, Yacooub Emilio Salem Reyes, gracias tesoro por brindarme noches de calma y tranquilidad. Tu me inspiraste y me diste fuerza para salir adelante. Te Amo.

A mi Amigo, Ing. José Ángel Posada, gracias por tu apoyo, tu paciencia y amabilidad, porque además de ser un excelente guía, siempre fuiste un gran amigo.

A mí Asesor, Ing. Carlos Bran, por tu paciencia y apoyo.

A mis grandes Amigos: Carlos Zambrano y José Salvador, porque siempre me brindaron su apoyo y estuvieron a mi lado a pesar de la distancia y las duras pruebas por las que hemos pasado.

Salem Emilio Salvador Salem Zablah

DEDICATORIA

A Dios:

Por permitirme llegar hasta aquí, y por enseñarme que al final cada sacrificio tiene su recompensa.

A mi Papa Benjamin y mi mama Rubidia:

Por darme la oportunidad de superarme, ya que gracias a su trabajo y esfuerzo me ayudaron a finalizar mi carrera. Gracias por confiar en mí.

A mi hermana:

Por el apoyo y la confianza que me brindo a lo largo de mi carrera.

A mi Abuelita Tere:

Por ser esa persona con la que puedo contar.

A mi novio Nelson: Por hacerme saber que las oportunidades que la vida nos presenta hay que aprovecharlas, gracias a su amor y compañía que espero sea para siempre.

A mis tías Claudia, Nelly, Aída: Por darme esos empujones que fueron necesarios para no desistir, y poner mas empeño en el logro de mis metas.

A mi prima Gaby y ahijado Marcos:

Por brindar alegría en mi vida.

A mis Amigos Patty y Salem:

Gracias por tener paciencia, porque a pesar de todo salimos adelante. Y pueden contar conmigo siempre.

Sinceramente,

Marcela

INDICE

INTRODUCCION.....	i
-------------------	---

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA.....	8
---------------------------------	---

I.1 ANTECEDENTES	8
I.2 PROBLEMA ACTUAL.....	9
I.3 JUSTIFICACION	12
I.5 OBJETIVOS	16
I.5.1 OBJETIVO GENERAL.....	16
I.5.2 OBJETIVOS ESPECÍFICOS.....	16
I.6 ALCANCES Y LIMITACIONES	17
I.6.1 ALCANCES.....	17
I.6.2 LIMITACIONES	18

CAPITULO II

MARCO TEORICO	22
---------------------	----

II.1 DEFINICIONES	22
II.1.1 INTERNET	22
II.1.2 ¿QUÉ ES UN SITIO WEB?	23
II.1.3 LENGUAJE DE MARCAS DE HIPERTEXTO (HTML)	24
II.1.4 WORLD WIDE WEB (WWW, WEB o W3)	24
II.1.5 BROWSER	25
II.1.6 PROTOCOLOS DE COMUNICACIÓN	25
II.1.6.1 PROTOCOLO IP	26
II.1.6.2 PROTOCOLO TCP/IP	26
II.1.7 EN LÍNEA	27
II.1.8 BASE DE DATOS	27
II.1.9 LINUX	28
II.1.10 JAVA	29
II.1.12 JAVA SCRIPT	30
II.1.13 ARQUITECTURA CLIENTE/SERVIDOR	31
II.1.14 SQL (LENGUAJE ESTRUCTURADO DE CONSULTAS)	32

CAPITULO III

ANALISIS Y DISEÑO DE LA INTERFAZ	33
II.1 ANALISIS DE LOS REQUERIMIENTOS	33
III.1.1 ANALISIS DE LAS ENCUESTAS	34
III.1.1.1 TABULACION DE ENCUESTAS DE DOCENTES	35
III.1.1.2 TABULACION DE ENCUESTAS ALUMNOS	45
III.2 REQUERIMIENTOS DEL SOFTWARE	53
III.3 REQUERIMIENTOS DEL HARDWARE.....	55
III.4 DIAGRAMAS DE FLUJO DE DATOS (DFD).....	56
III.4.1 DEFINICION Y NOTACION UTILIZADA EN UN DFD	56
III.4.2 DESCRIPCION DE LOS DIAGRAMAS DE FLUJO	58

CAPITULO V

ELEMENTOS PRINCIPALES DE LA INTERFAZ.....	59
IV. 1 ELEMENTOS PRINCIPALES	59
IV. 2 EL EDITOR.....	63
IV.3 EL QUIZ DE EJERCICIOS	65
IV.4 LEER COMENTARIOS	66
IV.5 PHPMYADMIN.....	66

CAPITULO VI

HERRAMIENTAS QUE PROPORCIONAN AYUDA PARA EL DESARROLLO DE SITIOS WEB	68
V.1 INTRODUCCION	68
V.2 PUBLISHER DE MSOFFICE	70
V.3 NUESTRO.ELSALVADOR.COM	71
V.4 FRONTPAGE.....	73
CONCLUSIONES	75
RECOMENDACIONES.....	79
VENTAJAS Y/ O BENEFICIOS	81
BIBLIOGRAFIA.....	82
GLOSARIO TECNICO	84
ANEXOS.....	87

INTRODUCCION

Un desafío que se enfrenta hoy en día es como proporcionar información eficiente y eficazmente, estando El Salvador inmerso dentro de un proceso de desarrollo las diferentes organizaciones (gubernamentales, empresariales, educativas, etc.) se ven forzadas a cambiar antiguos procedimientos y esquemas por tecnologías más avanzadas en el área de la informática, las cuales haciendo uso del hardware y software adecuados a sus necesidades, las proveen de altos estándares de competitividad y eficiencia.

Uno de los cambios más significativos tiene que ver con la educación, cada avance de desarrollo tecnológico abre posibilidades que pueden ser aprovechadas en un sistema educativo.

El presente documento contiene el desarrollo del trabajo de graduación que consiste en la "ELABORACION DE UNA INTERFAZ AUTOMATICA PARA LA CREACION DE CURSOS ACADEMICOS VIA WEB"

El cual se ha dividido en capítulos, los que se describen a continuación:

- **Capítulo I** define los objetivos general y específicos, alcances, limitaciones, antecedentes, definición del problema, justificación del proyecto, así mismo la solución propuesta y la metodología utilizada para recolectar la información necesaria.
- **Capítulo II** incluye las definiciones de términos comunes y técnicos, utilizados en el desarrollo del proyecto, para facilitar la comprensión del mismo.
- **Capítulo III** está dividido en dos etapas, primero el Análisis de Requerimientos, en el cual se hizo uso de ciertas técnicas de investigación como bibliografía de Internet, revistas, periódicos; encuestas y entrevistas para recolectar la información oportuna, las necesidades de los docentes y alumnos, los requerimientos de Software del producto, diagramas de flujo de datos para conocer como será el flujo de la información en el sistema de la interfaz. La segunda etapa incluye el diseño de la base de datos definido en el diagrama de entidad relación y el diccionario de datos.

- **Capítulo IV** describe de forma general, los elementos que comprenden el desarrollo de la interfaz, como el editor, los quiz de ejercicios, los mensajes por parte de los alumnos.

- **Capítulo V** presenta una pequeña investigación referente a otras herramientas que permiten la generación de Sitios Web, y que por sus funcionalidad, se parecen al de esta Interfaz.

Para finalizar se incluyen las conclusiones y recomendaciones así mismo algunas ventajas o beneficios de la aplicación en el desarrollo de este proyecto de tesis, proporcionándose también la bibliografía, glosario técnico y anexos, que amplían y respaldan la información presentada.

De igual forma se incluyen los manuales técnicos y de usuario; el manual técnico contiene la explicación de la estructura de la base de datos, los programas fuentes y otras especificaciones de instalación de la aplicación y el software utilizado para la elaboración y funcionamiento del mismo.

En el manual del usuario se explican las pantallas que forman parte de la aplicación, para que los docentes, se guíen cuando tengan una duda en el uso de la herramienta.-

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

I.1 ANTECEDENTES

En la actualidad nuestro país no cuenta con herramientas que proporcionen la creación de cursos en línea, solamente se hace uso de aquellas herramientas que permiten la generación de hojas Web, lo que para algunas personas resulta un poco difícil porque no conocen mucho sobre el diseño de páginas en Internet; así mismo también existen sitios dedicados a impartir diferentes cursos o clases conocidos como "Cursos en Línea", los cuales tienen la característica de ser elaborados por instituciones educativas como las universidades, que ofrecen a sus estudiantes la oportunidad de recibir clases vía Web.

En la sección de Internet del Diario de Hoy, se encontró un artículo referente a una herramienta que permite la elaboración de páginas Web, dicha herramienta está orientada a crear sitios para diferentes áreas como: industria, educación, comercio, etc., la cual está a disposición de aquellas personas que deseen colocar un sitio Web de forma gratuita, sin embargo esta es una aplicación para capturar información y diseñar páginas solamente, con datos teóricos y fotos, por lo que no se considera como una herramienta de creación de cursos en línea, ya que no permite la administración de los sitios elaborados porque no está diseñado para ello, generando poca interacción con el usuario, pues no permite modificaciones en tiempo real, y por ser una herramienta creada en el exterior no puede incorporarse esta opción, puesto que su código fuente no ha sido proporcionado para realizar estas mejoras a su funcionamiento.¹

Así mismo en la Prensa Gráfica, también se publicó otro artículo, el cual trataba sobre un tutor, llamado "TutorNet", la característica de este tutor es que está orientado al desarrollo de cursos académicos en línea, ideal para aquellos docentes que quieren incursionar en Internet, ya que permite generar clases de una manera muy sencilla al

¹ El Diario de hoy, Sección Negocios/Internet, publicado el 01 de Marzo de 2000.

utilizar el formulario que proporciona, el cual permite digitar lo que el docente necesita publicar en su página web, la desventaja es que no ofrece mucha interactividad con el usuario porque no permite incluir quiz de ejercicios o administrar la información que se almacena en las clases, ya que su diseño no puede modificarse, pues al igual que el anterior están diseñados a nivel de paquetes, y no son considerados como herramientas de elaboración y administración de materias en Internet, y aunque deja la siguiente inquietud: "es importante que se tenga en cuenta que el buen uso de la tecnología actual, permitirá reforzar los recursos de enseñanza mediante las facilidades que proporciona, y el docente al utilizarlas, para mejorar el aprendizaje de sus alumnos, ayuda no solamente a los ellos, sino a él mismo puesto que aprende más sobre la nueva tecnología, que hoy en día se ha vuelto indispensable"², no se puede asegurar que este sea implementado en nuestro país.

De igual forma están los cursos en línea de Cisco System³, a los cuales para tener acceso a ellos se debe tener un usuario y una contraseña.

Este sitio tiene en su página principal secciones las cuales al introducir el usuario y la contraseña asignada, permiten ingresar al sitio donde se presenta la lista de todas los cursos disponibles, y al elegir uno de ellos posteriormente se accesa a la información que dicha clase contiene.

También es importante mencionar que existen sitios que proporcionan cursos de forma gratuita, aunque su contenido sea poco académico.

I.2 PROBLEMA ACTUAL

Dentro de la Universidad los catedráticos encargados de impartir las materias en cada ciclo, llevan un control sobre los contenidos de sus clases de manera no automatizada, es decir el desarrollo de sus clases son plasmadas en hojas de papel, esta preparación de clases, al final de cada ciclo se convierte en un documento o informe que debe contener el desarrollo teórico de cada asignatura, incluyendo en el: temario, objetivos,

² La Prensa Gráfica, Sección Economía/ Internet, pág. 42, publicado el 26 de Abril de 2000.

³ <http://cisco.netacad.net>

metodología de evaluación, guías de laboratorio, etc., los cuales quedan como constancia del desarrollo de su cátedra.

Para aquellos docentes que laboran a tiempo completo en la Universidad puede decirse que no hay mayor contratiempo; mientras que para aquellos docentes o profesionales que imparten clases de una asignatura, solamente por Hora-Clase en la Universidad, se vuelve un poco complicado desarrollarlo, ya que tienen otras actividades que atender y no cuentan con el tiempo necesario para elaborar dicho informe de forma completa, es por eso que en algunas ocasiones no queda constancia de la metodología de enseñanza de estos docentes o profesionales que trabajaron para la Universidad una vez, perdiendo de esa forma un aporte para la educación en la Universidad; o que otros docentes puedan retomar esta información en beneficio propio y les sirva como base para desarrollar los temas que explicaran a sus alumnos en clase.

En cuanto al control que la Universidad Don Bosco, tiene actualmente para verificar que los planes de estudio para cada asignatura sean desarrollados de forma completa y que los objetivos se cumplan como están establecidos, lo hacen de forma manual, entendiéndose esto como la no aplicación de un sistema de administración por clase de cada asignatura desarrollada por el docente, en medios computarizados, por lo tanto se puede concluir que almacenan gran cantidad de papel.

Ahora bien para aquellos estudiantes que por algún motivo: como el trabajo, enfermedad, problemas personales, etc., dificultad para comprender en clases, que son tímidos para consultar directamente en el aula, que quieren reforzar sus clasesus notas, por mencionar algunos motivos de justificación, no poseen un medio donde puedan consultar el contenido teórico y/o práctico de las materias que cursa en el ciclo; más que de los apuntes de otro compañero de clase, o del libro texto indicado por el docente.

Concretamente, la Universidad Don Bosco necesita desarrollar un proyecto que facilite a los docentes crear aplicaciones que permitan incursionar en la nueva era de la tecnología, para dar mayor apoyo a los esquemas tradicionales de la educación. Y aunque algunos sepan utilizar una computadora, no significa que pueden elaborar

aplicaciones que generen procedimientos mediante el diseño o la programación de un listado de instrucciones para generar dicho proceso. Es por ello que se necesita de una aplicación que los docentes vean como una herramienta de edición común, donde ellos digiten sin problemas el contenido de su clase y que permita introducir animaciones, para hacerla llamativa, establecer exámenes de ejercicio solamente sin ninguna ponderación o que sus alumnos se comuniquen con él mediante un correo electrónico, y que facilite la administración de dichas clases, los accesos de los docentes y de los alumnos a la aplicación, por seguridad. Y que se aprovechen los recursos y facilidades que posee en la actualidad la Universidad Don Bosco.

1.3 JUSTIFICACION

Internet, esta siendo usada por muchas personas, empresas públicas o privadas e instituciones gubernamentales, entre otras, y las instituciones educativas no son la excepción, en tal sentido es preciso hacer notar que el avance tecnológico no solamente beneficia a las nuevas generaciones, sino también a aquellas personas que tengan el deseo de adquirir nuevos conocimientos para no irse desfasando por no conocer de los beneficios que ofrece hoy en día la computadora y desarrollar aplicaciones mecanizadas, diferentes al trabajo cotidiano que desarrollan. Demostrándose a ellos mismos que la enseñanza puede ir de la mano con los avances tecnológicos.

Es por ello que el desarrollo de esta aplicación se considera un valioso aporte, ya que al hacer uso de los recursos técnicos con los que la Universidad Don Bosco cuenta, proporcionan los medios y las facilidades para que tanto los docentes como los alumnos, puedan beneficiarse. Al desarrollar esta aplicación se tomo en cuenta que actualmente en la Universidad Don Bosco existe la necesidad de incorporar en sus servicios uno que permita a los alumnos encontrar sus clases en línea, y que al mismo tiempo proporcione a la Universidad un control para que los programas de estudio para cada asignatura, se desarrolle de la forma establecida, dejando de esta manera atrás la entrega de informes de asignaturas al final de cada ciclo.

La aplicación permite la creación y administración de cursos académicos, en donde su elemento principal es una interfaz que brinda al docente los pasos para la elaboración y colocación de los cursos en Internet, para digitar sus clases el docente hace uso de un Editor de texto, el cual permite hacer modificaciones con las propiedades de texto como cursiva, negrita, etc., la presentación de ejercicios que permite se contestarlos y obtener un puntaje no evaluado, y la comunicación con alumnos por medio de mensajes. Así los docentes que conocen poco sobre aplicaciones en computadoras, diseño de Cursos en Línea, creación de aplicaciones para el Web, puedan elaborar fácilmente sus clases, ya que la aplicación se considera como una herramienta de

trabajo que proporciona la ayuda y la guía necesaria para crear sus clases en un ambiente amigable y fácil de usar.

En cuanto a las herramientas o lenguajes de programación a utilizarse para desarrollar la interfaz, están el sistema operativo Linux por las siguientes razones:

- Licencia libre pues no tiene ningún costo,
- Soporte en línea pues tiene una página en Internet para bajar información y actualizaciones de las versiones nuevas de dicho sistema operativo,
- Estabilidad, refiriéndose esto a la seguridad en el sistema operativo,
- Proporciona el código fuente, para realizar cambios a dicho código según las necesidades del usuario,
- Manipulación en el orden de ejecución de los procesos, ya que permite detenerlo en el momento en que de algún problema.
- Requiere poco recurso en hardware.
- Posee un manejador de Bases de Datos que trabaja con sentencias SQL para manipular datos.
- Es compatible con el lenguaje de programación PHP3 el cual genera páginas en HTML y JavaScript para la manipulación de datos.

Mysql, por ser un manejador de bases de datos que aparte de manejar datos de tipo carácter, numéricos y de imágenes, es un servidor de bases de datos SQL multi-usuario y multitarea, que proporciona el almacenamiento de información de hasta 300 Mb por tabla en una Base de Datos.

I.4 SOLUCION PROPUESTA

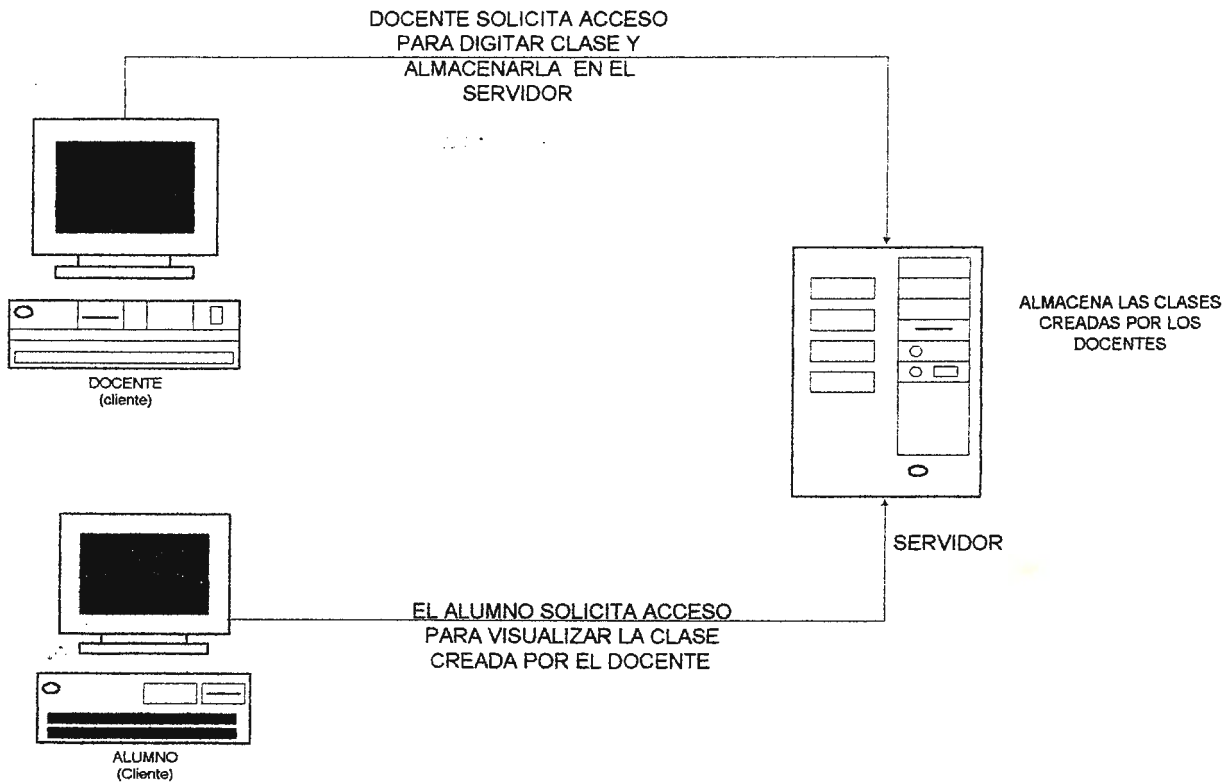
El proyecto propone desarrollar una aplicación que permita a los docentes diseñar sus clases en Internet. La cual proporciona la ayuda necesaria para que los maestros que conocen poco sobre la creación de sitios Web o programación HTML para el diseño de páginas Web, puedan elaborar un Curso en Línea, y para aquellos que si conocen de programación en Internet, puedan manipular las página que en la aplicación se elaboran.

La aplicación funciona bajo el esquema de Cliente/ Servidor. Levantando el Servidor con el sistema operativo Linux, por las razones explicadas anteriormente. El docente ingresa a través del cliente a la aplicación almacenada en el Servidor, mediante la asignación de un usuario y una contraseña, los cuales son autorizados por el administrador de la Base. La información de los cursos que se maneja en la aplicación es almacenada en una Base de Datos, con el objetivo de que queden a disposición de la Universidad Don Bosco para ser consultados, o actualizados por otros docentes en posteriores ocasiones, esta base de datos está en el servidor, y ha sido creada en MySql, porque trabaja bajo el ambiente del sistema operativo Linux y PHP, donde PHP es el lenguaje de programación que permite desarrollar funciones para poder manipular la información que se necesita almacenar en ella y que también interactúa con el lenguaje HTML para el diseño de la hojas Web.

Con la combinación de estos lenguajes de programación, se ha diseñado un Editor de texto, el cual permite elaborar una página Web con el contenido de la clase desarrollada por los docentes, así mismo permite incorporar ilustraciones como fotos, gráficos, dibujos, gif animados, sonidos, realidad virtual, etc., referentes a los temas desarrollados en los cursos.

Además esta aplicación permite que los alumnos tengan también acceso a los cursos creados por los docente, y al igual que ellos deben registrarse para que el administrador de la base les asigne un usuario y una contraseña, permitiendo de esta manera que los alumnos sean beneficiados al proporcionar las clases que son expuestas en los salones de clase y que por algun motivo no pudieron asistir.

Gráficamente la aplicación funciona de la siguiente manera.



I.5 OBJETIVOS

I.5.1 OBJETIVO GENERAL

Diseñar y desarrollar una herramienta que permita la creación y administración de cursos académicos para ser colocados en Internet.

I.5.2 OBJETIVOS ESPECÍFICOS

- Diseñar una Base de Datos, para almacenar y administrar la información de los cursos académicos que el docente elabora en papel, así como desarrollar mantenimientos para adicionar, modificar, eliminar y consultar dicha información con el fin de mejorar el rendimiento y administrar eficientemente la base de datos.
- Elaborar métodos de seguridad, para controlar el acceso a la base de datos, minimizando el riesgo que la información pueda ser modificada por personas no autorizadas, tanto para los docentes como para los alumnos.
- Diseñar una interfaz de fácil uso, que permita la administración y creación de cursos académicos.
- Elaborar una plantilla de edición, que facilite al docente la creación de los cursos, que permita digitar, modificar y adicionar diferentes propiedades del texto y cada uno de los objetos que el docente seleccione.

I.6 ALCANCES Y LIMITACIONES

I.6.1 ALCANCES

- La herramienta trabaja con un listado de todas las materias a impartirse en el ciclo vigente con el objeto de que sean escogidas para crear la clase en el Web.
- Utilizando MySql como Administrador de Bases de Datos, que permite manipular datos de tipo texto, numéricos e imágenes, se diseña y crea una base de datos la cual almacena la información de las clases que impartirán los docentes.
- Contiene los mantenimientos que permiten adicionar, modificar, consultar o eliminar cursos académicos, administrando de esa manera la información almacenada en la base de datos, y mejorar el rendimiento de la misma, para evitar que el servidor contenga información innecesaria, lo que no permitiría su funcionabilidad adecuada.
- Proporciona métodos de seguridad, con el fin de proteger la información que los docentes almacenan en la base de datos, mediante la asignación de un usuario y una contraseña, a cada una de las personas involucradas en el uso de la herramienta, para la creación de los cursos académicos. Para lo cual se deberá llenar un formulario, pequeño y sencillo, que permite recolectar los datos personales de los usuarios que tendrán acceso a la Interfaz, el cual estará bajo responsabilidad del administrador de la base de datos.
- La Interfaz incluye un Editor de texto que facilita la generación de la clase al digitar la información teórica de la clase, así mismo incluye las opciones de evaluación-ejercicios o quiz de ejercicios, para colocar un listado de preguntas referentes al tema desarrollado.

- El uso de la aplicación esta orientado para docentes y alumnos de la Universidad don Bosco, exclusivamente, la cual esta situada en el municipio de Soyapango del departamento de San Salvador.
- Este asistente proporciona una alternativa de solución, únicamente, al problema de controlar a los docentes en el desarrollo de los programas de estudio para cada materia que imparten dentro de la Universidad Don Bosco y también para aquellos alumnos que no poseen otra forma de recuperar una clase perdida.
- Al finalizar el desarrollo del asistente se presenta un caso práctico orientado a un tema específico de la materia de Física

1.6.2 LIMITACIONES

- La creación de fórmulas está limitada por la tabla de Símbolos ASCII⁴ del sistema operativo.
- La creación de la clase está limitada por las propiedades de presentación del texto (tipo de letra, negrita, justificación de párrafo, tablas, etc) que posee el editor de la interfaz.
- La funcionabilidad de la interfaz ha sido ejecutada y probada en una Intranet pequeña, compuesta por un Servidor Linux y dos clientes con Windows 98, y debido a esto, la carga de la aplicación en el Browser puede variar en rapidez cuando sea ejecutada en Internet, dependiendo de los objetos que se incorporen a la clase.

- Esta aplicación está elaborada bajo la plataforma del sistema operativo Linux, por lo tanto su implementación en el Servidor Web de la universidad que funciona en Windows NT, queda bajo responsabilidad de las personas encargadas de esta área.
- El uso de la Interfaz, para la creación de Cursos en Línea, está diseñado exclusivamente para el personal docente de la Universidad Don Bosco, y de igual forma los alumnos de esta institución se beneficiaran con esta aplicación ya que solamente ellos tendrán acceso a las Clases en Línea.

1.7 METODOLOGIA DE LA INVESTIGACIÓN.

Para desarrollar el presente trabajo de graduación se tomó como parte fundamental el análisis de los requerimientos (necesidades) de los usuarios, la obtención de esta información se realizó en dos fases:

- Investigativa

Bibliografía

Esta se refiere a la consulta de información escrita, la cual ayuda al desarrollo del proyecto.

Para ello se buscó información sobre definiciones de interfaces, cursos en línea (Online), también analizaron los lenguajes de programación de tal manera que nos permitiera determinar cual era más factible para el desarrollo de nuestro proyecto.

Información en Internet

Esta ha sido la principal fuente de información, puesto que el tema de cursos en línea, no es muy conocido y no se encuentra bibliografía que haga referencia a este tema, a través de Internet se obtuvo información bastante actualizada sobre el tema, así como también los manuales de Linux, Mysql, Php3 y acceder a ciertos cursos en línea, que se encuentran a disposición de los usuarios en forma gratuita.

- Análisis de Requerimientos

En esta fase se elaboraron los procedimientos que permitieron recolectar los requerimientos del usuario, estos procedimientos fueron las encuestas y las entrevistas.

Las encuestas fueron orientadas a los docentes y los alumnos, con el fin de detectar el conocimiento, uso e importancia que tiene Internet hoy en día, para ellos, estas fueron tabuladas reflejando así las necesidades existentes, definiendo con ello los requerimientos del usuario.

Las entrevistas están orientadas a los docentes, para determinar el procedimiento que realizan en la elaboración del documento que presenta el detalle del contenido de las asignaturas que cada uno imparte.

- Elaboración de Pruebas

- Prácticas Experimentales

Para desarrollar este proyecto, se formó un laboratorio que funciona bajo el esquema de cliente – servidor, de manera que el cliente visualice las clases colocadas en línea.

CAPITULO II

MARCO TEORICO

II.1 DEFINICIONES

II.1.1 INTERNET⁵

Es la red de redes. Nacida como experimento del ministerio de defensa americano, conoce su difusión más amplia en el ámbito científico-universitario.

Desde el punto de vista técnico, Internet es un gran conjunto de redes de ordenadores interconectadas (la mayor red mundial). Desde otro punto de vista, Internet es un fenómeno sociocultural. Un usuario desde su consola, tiene acceso a la mayor fuente de información que existe.

En cuanto a funcionamiento interno, Internet no se ajusta a ningún tipo de ordenador, tipo de red, tecnología de conexión y medios físicos empleados.

Internet no tiene una autoridad central, es descentralizada. Cada red mantiene su independencia y se une cooperativamente al resto respetando una serie de normas de interconexión. La familia de protocolos TCP/IP es la encargada de aglutinar esta diversidad de redes.

A principios de 1.992 fué creada la Internet Society (ISOC)⁶. Se trata de una sociedad profesional sin ánimo de lucro, formada por organizaciones e individuos de todos los sectores involucrados de una u otra forma en la construcción de Internet (usuarios, proveedores, fabricantes de equipos, administradores, etc..). El principal objetivo es fomentar el crecimiento de la Internet en todos sus aspectos (número de usuarios, nuevas aplicaciones, infraestructuras, etc..).

⁵ <http://www.discapnet.es/graficos/documentacion/glosario/index.asp>. Glosario

⁶ <http://info.isoc.org/> Sociedad de Internet

II.1.2 ¿QUÉ ES UN SITIO WEB? ^{or}

Es un conjunto de archivos escritos en lenguaje de programación HTML, que están alojados en el servidor (computadora) que posee una dirección IP, es decir que es reconocido desde cualquier punto de Internet. Dentro de estos archivos es posible colocar instrucciones que hagan conexiones con otros archivos o sitios, los cuales pueden ser creados por la misma persona, o por otros que ya los hayan colocado en el Web. Un archivo HTML, en un sitio Web puede incluir, texto, gráficos, sonido, animaciones, videos, etc.

Las conexiones entre archivos HTML se realizan a través de enlaces(hiperlinks), que contienen:

- a) Un texto o imagen que se ve en el Browser del Web, con diferente formato de las demás imágenes o palabras del documento, indicando una redirección de búsquedas.
- b) Una parte oculta para el usuario, que apunta a la dirección IP del documento al cual se hace referencia.

En general el proceso que se sigue en la redirección por medio de los enlaces hipertexto, es el siguiente:

- Al cargar un documento Web dentro de un Browser del Web, el usuario puede seleccionar un enlace hacia otro sitio del documento.
- El cliente Web (Browser) se conecta a una computadora identificada por medio de una dirección IP, que funciona como un servidor.
- Dicho servidor responde a la solicitud enviando un documento hacia el Browser del cliente.
- El servidor sigue respondiendo ante cualquier acción que se realice con dicho documento, o por la presentación de imágenes en movimiento o sonidos dentro de ésta.

El protocolo que permite establecer un lenguaje común entre cliente y servidores Web es llamado Protocolo de Transferencia de Hipertexto.

II.1.3 LENGUAJE DE MARCAS DE HIPERTEXTO (HTML) ^{er}

Es el lenguaje de hipertexto estándar para W.W.W. El cual sigue las convenciones de formato generalizado para el lenguaje de marcado.

Los archivos que utiliza como fuente son simples archivos de texto ASCII, incluyendo códigos de formato y enlaces con otros documentos, de tal manera que para crearlos puede utilizarse cualquier editor de texto VI (UNIX.), Edit (Dos), Notepad (Windows), etc., o bien un procesador de palabras como: MS Word, WordPerfect, etc., dándole al archivo la extensión. Htm o.html.

Para poder visualizar el archivo Html, debe utilizarse un Browser como Netscape, NCSA Mosaic, Lynx, MacWeb, etc.

Html también soporta la creación de formas interactivas con el usuario, imágenes en movimiento, nuevos formatos de estilos de texto, así como también direcciones para el envío de correo electrónico.

II.1.4 WORLD WIDE WEB (WWW, WEB o W3) ^{er}

Su traducción literal es: Gran telaraña Mundial

Es un servicio de Internet que consiste en una inmensa red formada por pantallas gráficas de información ubicadas en computadoras distribuidos en todo el planeta.

Es el recurso más usado dentro de Internet, que ofrece un ambiente gráfico de navegación. Compuesto por documentos de distintos tipos, compatibles con un Browser de Web, el cual se utiliza para visualización de los mismos. Dichos documentos pueden editarse utilizando un lenguaje de hipertexto especial, llamado HTML.

Permite obtener información de cualquier tema, tal como conferencias, eventos, trabajos, agencias de gobierno, museos, etc.

^{er} " Interfaz CGI para Servidores Web y Sistemas de Administración de Bases de Datos " TESIS UCA, Licenciatura en Computación, Capítulo II.

II.1.5 BROWSER ^{er}

Es un programa (como Nescape). Mosaic, Microsoft Internet Explorer) que permite ver la información en el W.W.W. en un formato gráfico. El intérprete estándar usado para Browser del web es el HTML (Hypertext Markup Lenguaje). Algunos Browser también interpretan VRML (Virtual Reality Markup Lenguaje) y/o Java, el cual permite tener algunas características para la animación de paginas web.

Las características de Browser son:

Pueden interactuar de diversas formas, tales que les permiten comunicarse con todos los servidores (Gopher, FTP, Web), actuando como clientes y utilizando los protocolos adecuados.

- a) Emplean una interfaz gráfica con el usuario.
- b) Permiten hacer referencias hacia información en hipertexto o hipermedia. De esta forma, cualquier palabra, frase o imagen, puede funcionar como enlace("Hotlink") hacia cualquier otra información, sin necesidad de utilizar directamente los comandos de UNIX, como se hacía en años anteriores.

II.1.6 PROTOCOLOS DE COMUNICACIÓN ^{er}

Es el lenguaje estándar que se utiliza en Internet para el intercambio de información a través de la red.

Conjunto estricto de reglas o procedimientos que se requieran para iniciar y mantener las comunicaciones, son los que hacen posible el intercambio de información, después de establecer una llamada a través de un canal informativo.

Cuando se envía un mensaje a través de Internet, éste es dividido en un conjunto de "Paquetes", los cuales viajan por distintas rutas, desde la computadora origen hasta

llegar a su destino. Para la realización de esta tarea se hace necesario contar con protocolos que permitan establecer la comunicación y transferencia de dichos paquetes a través de la red.

El sistema de protocolos que fue desarrollado como producto de las primeras investigaciones realizadas por el Departamento de Defensa de los Estados Unidos, es que se llegó a conocer como TCP/IP, después de que se desarrollaron los dos protocolos iniciales: el Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP)

II.1.6.1 PROTOCOLO IP ^{er}

Es el responsable de mover paquetes, de datos desde un nodo a otro, guiándose cada paquete mediante el uso de una dirección de cuatro bytes, que funciona en las máquinas conectadas a la red.

II.1.6.2 PROTOCOLO TCP/IP ^{er}

Es un conjunto de protocolos desarrollado para permitir a las computadoras compartir recursos a través de la red, de tal manera que establecen una comunicación constante entre ellas. Por ello, este es un conjunto básico para la comunicación y transmisión de datos en el Internet.

^{er} " Interfaz CGI para Servidores Web y Sistemas de Administración de Bases de Datos " TESIS UCA, Licenciatura en Computación, Capítulo II.

II.1.6.3 PROTOCOLO HTTP ^{er}

Ha estado en uso en el WWW, desde 1990, como un protocolo genérico orientado a objetos, que puede ser usado para distintas tareas tales como servidores de aplicaciones y sistemas de control de distribución de información. Una característica de HTTP es que permite al sistema cargarse independientemente de que los datos se estén transfiriendo.

II.1.7 EN LÍNEA ⁷

Conectado a Internet, donde pueden encontrarse oportunidades generadas por servicios de información electrónica y aplicaciones para el desarrollo de un negocio. Explotando creativamente las fuentes ya existentes de información en áreas de interés, para transmitir su valor inherente a través de un medio centralizado y establecer una red global de información.

Con lo cual se puede cubrir la demanda del público y los medios a través de una red automatizada de servicios, logrando incrementar el valor comercial de la información permitiendo el fácil acceso y cubriendo demandas, para brindar un servicio de información universal continuo que sea accesible independientemente de su ubicación geográfica, mediante una vía de comunicación eficiente e interactiva, en el instante.

II.1.8 BASE DE DATOS ^{er}

Es básicamente un sistema computarizado de almacenamiento en registros, utilizado con el propósito de darle mantenimiento a la información y hacerla disponible ante cualquier demanda.

Los componentes de un sistema de bases de datos son:

⁷ <http://www.comunicacionesonline.com/ponline.html> ¿Por qué Online?

^{er} " Interfaz CGI para Servidores Web y Sistemas de Administración de Bases de Datos " TESIS UCA, Licenciatura en Computación, Capítulo II.

1. Datos, que son alojados totalmente en una base de datos para brindar una mayor seguridad, además de ser compartidos e integrados con una mayor facilidad; se entiende por 'compartir' el hecho de que distintos usuarios pueden hacer uso de manera simultanea de los datos, para distintos propósitos, y, por 'integración' se entenderá la unificación de distintos archivos de datos que no poseen redundancia.
2. Hardware, que son los volúmenes de almacenamiento en los cuales reside la base de datos, de manera física, y también son dispositivos de entrada/salida.
3. Software, que se encuentra entre la base de datos física y el usuario; en definitiva, el software es el administrador de la base de datos (DB manager), más usualmente conocido como sistema de administración de bases de datos (DBMS).
4. Usuarios, que pueden ser programadores de aplicaciones, usuarios finales o administradores de bases de datos, que de una u otra forma se mantienen en contacto con la base de datos, realizando cualquier operación relacionada con la información que ésta contenga.

II.1.9 LINUX ⁸

Linux fue creado originalmente por Linus Torvald en la Universidad de Helsinki en Finlandia. Pero ha continuado su desarrollado con la ayuda de muchos otros programadores a través de Internet.

Linux es una versión de UNIX libremente distribuible e independiente, para plataformas con maquinas x86, Motorola 68k, Digital Alpha, Sparc, Mips y Motorola Power PC. Es una implantación de la especificación POSIX con la cual cumplen todas las verdaderas versiones de Unix. El núcleo de Linux no usa código de AT&T o de cualquier otra fuente propietaria, la mayoría de los programas disponible para Linux son desarrollados por el proyecto GNU de la Free Software Foundation (Fundación de programas Libres). Este soporta un amplio espectro de aplicaciones o paquetes de programación tales, como X Window, Emacs, redes de datos bajo protocolos TCP/IP (incluyendo SLIP,

⁸ <http://www.internet.ve/asic/glinux.html> ¿Qué es Linux?

PPP, ISDN). Mucha gente ha corrido benchmarks en sistemas Linux 80486 consiguiendo a esto comparable con estaciones de rango-medio de Sun y Digital.

Linux esta disponible en Internet en cientos de servidores ftp y de varios distribuidores en discos flexibles o CD-ROM, alguno de estos son Caldera, Debian, Slakware, Red Hat, Suse, entre otros. Uno de los sitios de ftp más famoso es el <ftp://sunsite.unc.edu/pub/Linux/distributions>, con decenas de servidores espejos alrededor del mundo.

El núcleo de Linux esta legalmente protegido por la Licencia Publica GNU (GPL de sus siglas en ingles), es usualmente empaquetado con varios ejecutables que comprenden un sistema operativo Unix funcional. Estos paquetes de programas son llamados "distribuciones" y vienen en muchos tamaños y arreglos.

II.1.10 **JAVA** ^{er}

Java es un lenguaje orientado a objetos y desarrollado por Sun Microsystem. Comparte similitudes con C, C++ y *Objective C*. El principal objetivo de JAVA fue hacer un lenguaje que fuera capaz de ser ejecutado de una forma segura a través de Internet. Es tanto un compilador como un lenguaje interprete. El código fuente Java es convertido en instrucciones binarias simples, y compilado con formato universal. El compilador realiza todas las actividades de un procesador real en un ambiente virtual mas seguro. Es decir ejecuta instrucciones, crea y manipula información, carga y hace referencia a bloques de códigos nuevos.

^{er} " Interfaz CGI para Servidores Web y Sistemas de Administración de Bases de Datos " TESIS UCA, Licenciatura en Computación, Capítulo II.

II.1.11 APLICACIONES JAVA ^{er}

Los programadores pueden desarrollar pequeñas aplicaciones, las cuales permiten tener sitios Web con una gran funcionalidad en cuanto a: Animación, actualización en vivo, interacción bidireccional y más.

Al integrarse en una página Web, las aplicaciones de Java tiene acceso a:

- Recreación de gráficos expertos
- Interacción en tiempo real con los usuarios
- Actualización en vivo de la información
- Interacción instantánea con los servidores a través de la red

II.1.12 JAVA SCRIPT ^{er}

Es un lenguaje muy poderoso y especialmente diseñado para la creación de escritos, que se alojan dentro de un documento HTML. Dicho lenguaje es propiedad de Netscape.

Es un API programable que permite crear escrito de eventos, objetos y acciones, bajo cualquier plataforma. Gracias a que JavaScript es parte de la conexión en vivo se puede usar para crear interacciones entre documento HTML, Plug-ins (aplicaciones que corren dentro del Browser del Web) y Java.

Es un lenguaje de escritos compactos, basado en objetos, para el desarrollo de aplicaciones Internet Cliente/Servidor. Las sentencias JavaScript se reconocen o responde ante eventos, pueden ser introducidas directamente en una página Web. Por ejemplo, se puede escribir una función JavaScript que verifique a la correcta entrada de datos a una forma sin necesidad de transmisión de datos a través de una red. Así, una pagina HTML con código JavaScript puede interpretar el texto introducido y alertar al usuario si el dato es invalido.

II.1.13 ARQUITECTURA CLIENTE/SERVIDOR ^{et}

Es una arquitectura computacional que involucra procesos de clientes que se encuentran requiriendo servicios de procesos de servidor.

..Cliente/Servidor es el concepto computacional que viene a ser la extensión lógica de la programación modular, la cual asume fundamentalmente la separación de grandes piezas de software, en partes más pequeñas llamadas "módulos", creando la posibilidad de obtener un desarrollo más fácil y darle un mejor mantenimiento. El proceso cliente/servidor reconoce que estos módulos no necesitan ser ejecutados dentro del mismo espacio de memoria, de tal manera que al utilizar esta arquitectura, el módulo que realiza la llamada se convierte en el "Cliente" (que es quien hace la requisición de un servicio), y el módulo que es llamado se convierte en el "Servidor" (que es el que provee el servicio). Como se muestra en la figura 1

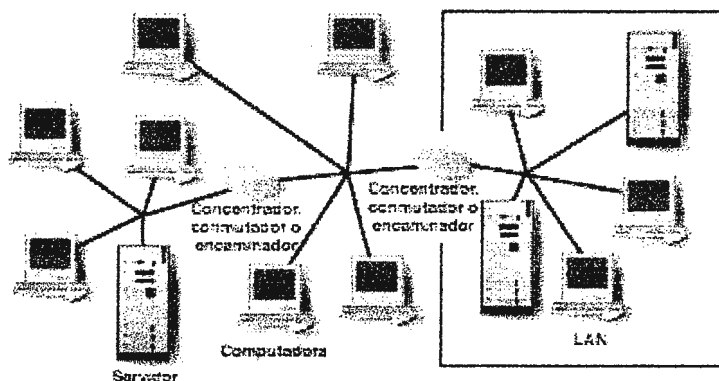


FIGURA 1

II.1.14 SQL (Lenguaje Estructurado de Consultas) ^{er}

Es un sublenguaje estándar para acceder y manipular la información que se encuentra dentro de los sistemas de bases de datos.

La mayoría de dichos sistemas que se usan en la actualidad, como Oracle o Sybase, soporta sentencias SQL.

Con SQL se puede realizar ciertas acciones sobre la información tales como: Crear base de datos, acceder a la información dentro de las bases de datos, para realizar consultas específicas, actualizar la información dentro de la base de datos, borrar datos específicos de la base.

^{er} " Interfaz CGI para Servidores Web y Sistemas de Administración de Bases de Datos " TESIS UCA, Licenciatura en Computación, Capítulo II.

CAPITULO III

ANALISIS Y DISEÑO DE LA INTERFAZ

II.1 ANALISIS DE LOS REQUERIMIENTOS

La tarea del análisis de requerimientos es un proceso de descubrimiento, refinamiento, modelado y especificación de las necesidades de los usuarios.

Un requerimiento es una característica que debe incluirse en un nuevo sistema. Esta puede ser la inclusión de determinada forma para capturar o procesar datos, producir información, controlar una actividad de la empresa, etc.

El análisis y la especificación de los requisitos puede parecer una tarea relativamente sencilla; los requerimientos se definen tomando en cuenta las necesidades de los usuarios, que permiten desarrollar un software de mayor calidad.

Para lograr detectar los requerimientos de cualquier problema, se debe recurrir a diferentes metodologías para formular y priorizar las necesidades de los usuarios finales.

En el análisis de requerimientos, del proyecto, se elaboraron encuestas (**Anexo 1 y Anexo 2**), para ser resueltas por docentes y alumnos de la Universidad Don Bosco, y determinar con ello las necesidades de mayor importancia para los usuarios, tanto de la herramienta para crear las clases en línea, como de los beneficiarios directos de esta aplicación (los alumnos), aunque es importante mencionar que se pasaron encuestas en otras universidades para verificar que no solo en la universidad Don Bosco, es importante desarrollar proyectos como este, sino también en otras universidades del país, ya que esto beneficiaría mucho en la enseñanza a nivel superior.

La encuesta para docentes permitió detectar

- El uso que los docentes le dan a Internet
- El conocimiento que tienen sobre Cursos en Línea y que piensan de ellos
- Los tipos de herramientas de computación que conocen
- Los tipos de herramientas de computación sabe usar
- La utilidad que tiene Internet para ellos en su labor educativa

- Si saben diseñar Clases en Internet
- Las dificultades que tiene para elaborar una clase en Internet
- Si existiera una herramienta que les ayude a elaborar su clase en Internet, la usarían o no

Para los alumnos la encuesta permitió encontrar

- Si tienen o no acceso a Internet
- Desde donde accesan a Internet
- Que dificultades tienen para acceder a Internet
- Si la universidad les proporciona acceso a Internet
- Como es el servicio que les proporciona la Universidad, para acceder a Internet.
- Si conocen lo que son cursos en línea
- Que les parecería tener un curso en línea en la Universidad

Para la resolución de las encuestas a los docentes se tomo una muestra de 35 de la Universidad Don Bosco y 60 de distintas universidades del área metropolitana de San Salvador; para los alumnos la muestra fue tomada en la Universidad Don Bosco a un total de 150.

III.1.1 ANALISIS DE LAS ENCUESTAS

Para desarrollar el análisis de las encuestas se procedió de la siguiente manera, por cada pregunta se efectuó:

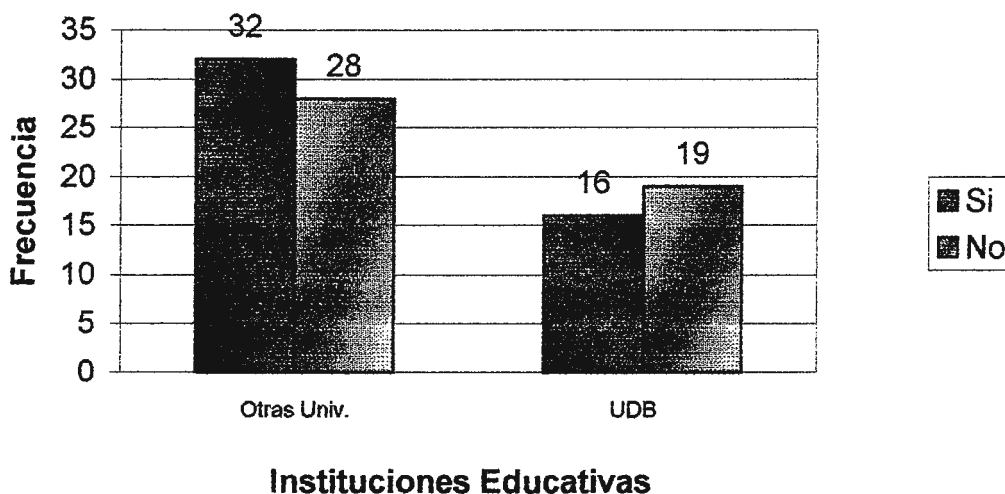
- Objetivo de la pregunta
- Pregunta efectuada
- Gráfico
- Hallazgo

III.1.1.1 TABULACION DE ENCUESTAS DE DOCENTES

Pregunta: ¿ Tiene acceso a Internet desde su casa?

Objetivo: Determinar si los docentes tiene acceso a Internet desde sus casas, ya que la mayoría de ellos desarrolla sus clases, fuera de las instalaciones educativas.

	Otras Univ.	Porcentaje	UDB	Porcentaje
Si	32	53%	16	46%
No	28	47%	19	54%
	60	100%	35	100%



Hallazgo: Con esta pregunta se determina que la mayoría de los docentes no accesan a Internet desde sus casas, por diferentes motivos. La mayoría opina que el costo de acceso, por línea telefónica es bastante elevado, y esa es la razón por la cual no utilizan el Internet, otros opinan que no les interesa Internet, porque simplemente no es su rama y como no pueden desarrollar aplicaciones para el Web, no les llama la atención.

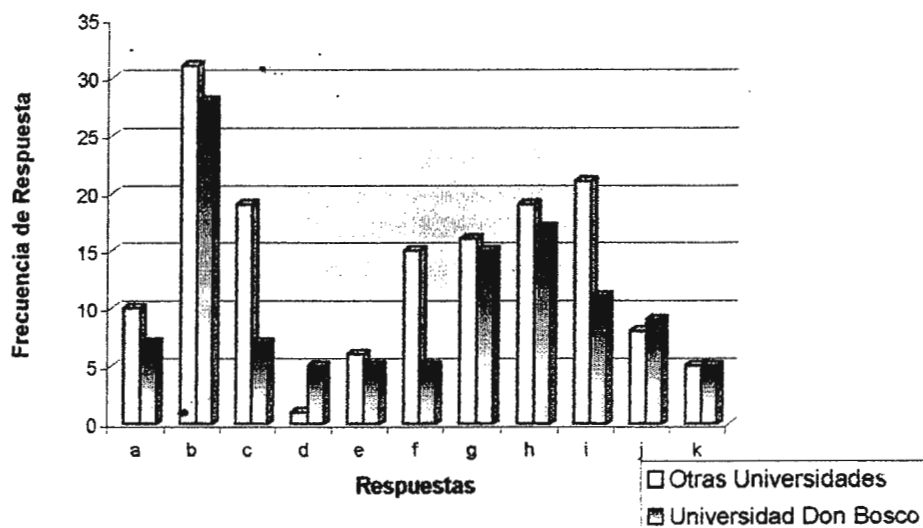
Pero los docentes que si usan Internet, ven a esta como una ventaja en la comunicación y como una facilidad en la adquisición de nuevas herramientas para su desarrollo propio, ya que les permite obtener información más reciente e innovadora, y el desarrollo de sus clases no se limita al contenido de los libros de texto.

• **PREGUNTA** ¿Qué uso le da actualmente a Internet?

Objetivo

Conocer el uso que los docentes le dan a Internet, para medir la factibilidad de elaborar sus clases y ponerlas en línea, a disposición de los alumnos.

	Otras Univ.	Porcentaje	UDB	Porcentaje
a. Búsqueda de información de entretenimiento	10	7%	7	6%
b. Búsqueda de información en general	31	21%	28	25%
c. Comunicación vía correo electrónico	19	13%	7	6%
d. Platicar en el Chat	1	1%	5	4%
e. Transacciones Bancarias o Económicas	6	4%	5	4%
f. Comercio electrónico(compra de libros, CD's de Software, música, etc.)	15	10%	5	4%
g. Actualizaciones de Software (Bajar Programas)	16	11%	15	13%
h. Consultas a Bibliotecas	19	13%	17	15%
i. Lecturas de revistas y periódicos	21	14%	11	10%
j. Foros de comunicación y/ o consulta	8	5%	9	8%
k.Otros	5	3%	5	4%
TOTALES	151	100%	114	100%



Hallazgo

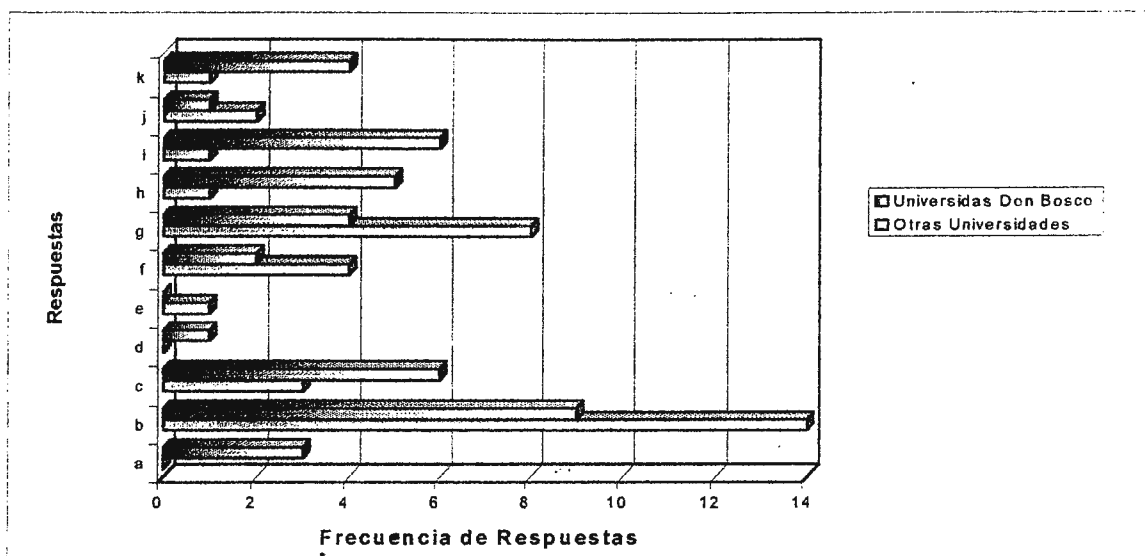
El 21% de los docentes en otros centros educativos, utilizan Internet, para la búsqueda de información en general, mientras que en la Universidad Don Bosco el 25% corresponde a la búsqueda de información en general. Esto indica que los docentes tienen acceso, lo utilizan y conocen Internet.

PREGUNTA ¿ De los usos antes mencionados, cuáles considera que apoyan el proceso de aprendizaje-enseñanza?

Objetivo

Determinar si el docente considera que Internet, le proporciona algún tipo de ayuda al proceso de aprendizaje-enseñanza

	Otras Univ.	Porcentaje	UDB	Porcentaje
a. Búsqueda de información de entretenimiento	0	0%	3	3%
b. Búsqueda de información en general	34	27%	35	33%
c. Comunicación vía correo electrónico	26	21%	15	14%
d. Platicar en el Chat	0	0%	2	2%
e. Transacciones Bancarias o Económicas	2	2%	0	0%
f. Comercio electrónico(compra de libros, CD's de Software, música, etc.)	9	7%	11	10%
g. Actualizaciones de Software (Bajar Programas)	19	15%	11	10%
h. Consultas a Bibliotecas	16	13%	9	8%
i. Lecturas de revistas y periódicos	5	4%	10	9%
k. Foros de comunicación y/ o consulta	9	7%	3	3%
Otros	5	4%	8	7%
TOTALES	125	100%	107	100%



Hallazgo

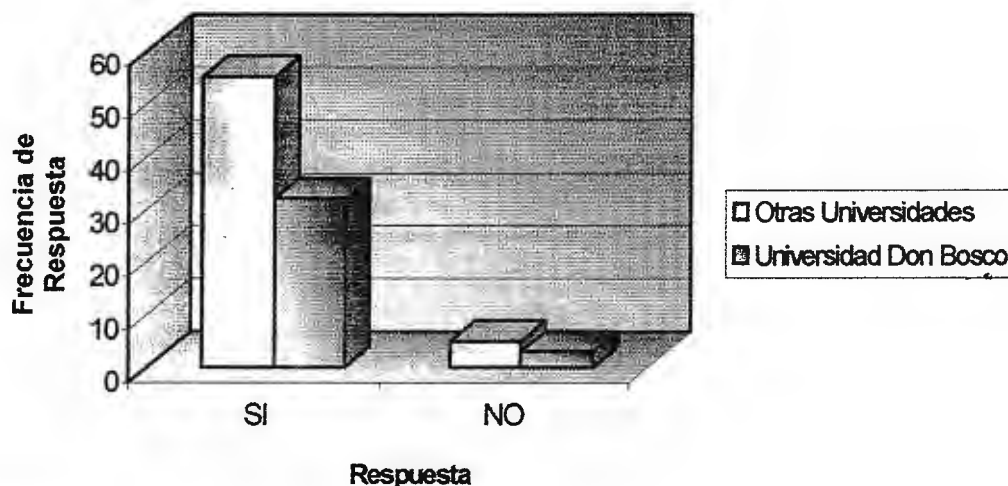
Se detectó que el 27% de los docentes opinan que la búsqueda de información es la que más apoya el proceso ya que con ello los alumnos pueden adquirir conocimientos más actualizados, de manera que fortalezcan su proceso de aprendizaje, en segundo plano consideran que la comunicación vía correo electrónico también aporta una ayuda, ya que al establecer contacto con personas de otros países, pueden también adquirir información más actualizada, al igual que las consultas a bibliotecas, pues ahí los alumnos pueden encontrar la bibliografía de una forma mucho más rápida.

PREGUNTA ¿Considera que los Curso en Internet mejoran el aprendizaje de los alumnos?

Objetivo

Saber si los docentes consideran que los Curso en Internet mejoran el aprendizaje de los alumnos.

	Otras Univ.	Porcentaje	UDB	Porcentaje
SI	55	92%	32	91%
NO	5	8%	3	9%
TOTALES	60	100%	35	100%



Hallazgo

El 92% de los docentes de otras universidades y en la Universidad Don Bosco el 91%, opinan que los cursos en Internet Sí contribuyen en el aprendizaje de los estudiantes, ya que consideran que con ello se les motiva pues las clases son más dinámicas, mientras que uno de cada diez docentes creen que los cursos en Internet evitarían que dieran la clase personalmente.

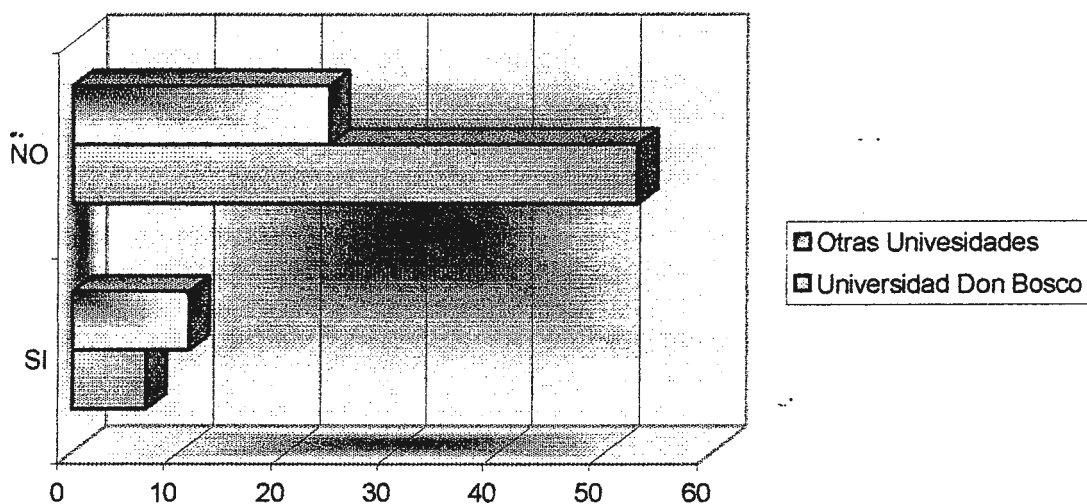
Aunque la minoría opina que lo único que les facilitarían estos cursos es ya no impartir clases.

Pregunta ¿ Puede Ud. Diseñar Cursos en Internet?

Objetivo

Determinar si los docentes pueden o no elaborar cursos en Internet

	Otras Univ.	Porcentaje	UDB	Porcentaje
SÍ	7	12%	11	31%
NO	53	88%	24	69%
	60	100%	35	100%



Hallazgo

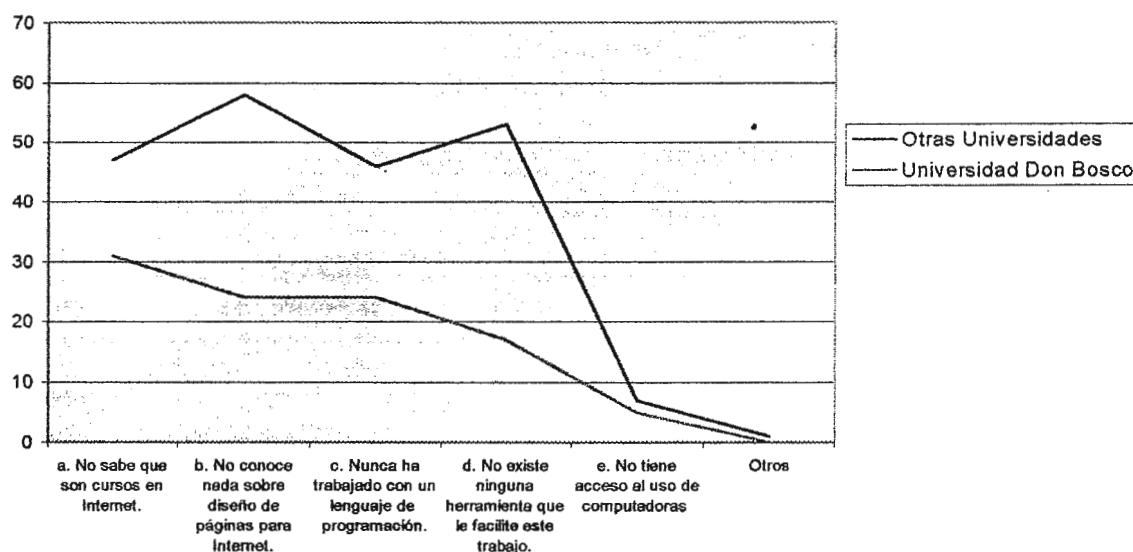
El 69% de los docentes de la Universidad Don Bosco, no saben diseñar cursos en Internet, en comparación a otras universidades que es el 88% de docentes que no pueden elaborar una clase en Internet, mientras que el 31% si pueden pues en su mayoría son docentes de la carrera de Ing. En computación, y son pocos los docentes de otras carreras, mientras que en otras universidades son pocos los docentes que pueden.

PREGUNTA ¿Cuál es la dificultad?

Objetivo

Determinar las dificultades que tienen los docentes para diseñar y elaborar una clase en Internet.

	Otras Univ.	Porcentaje	UDB	Porcentaje
a. No sabe que son cursos en Internet.	47	22%	31	31%
b. No conoce nada sobre diseño de páginas para Internet.	58	27%	24	24%
c. Nunca ha trabajado con un lenguaje de programación.	46	22%	24	24%
d. No existe ninguna herramienta que le facilite este trabajo.	53	25%	17	17%
e. No tiene acceso al uso de computadoras	7	3%	5	5%
Otros	1	0%	0	0%
TOTAL	212	100%	101	100%



Hallazgo

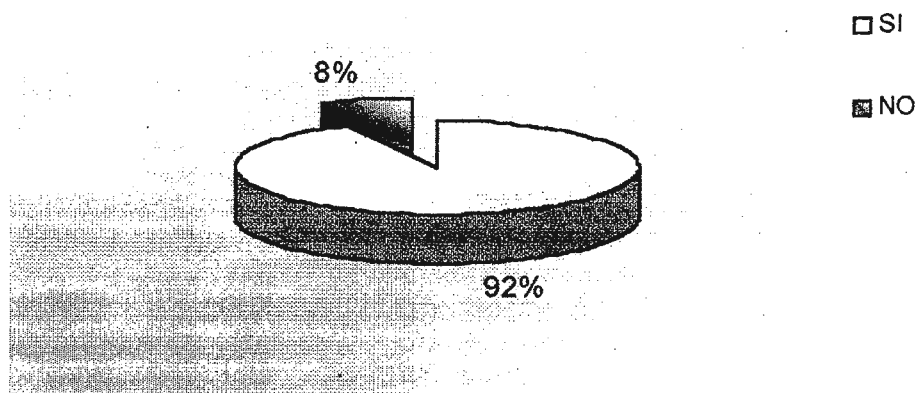
Esta pregunta determina en forma general, que los docentes en su mayoría, no saben que son Cursos Online, y que no han trabajado con lo que son lenguajes de computación, a diferencia de los que se encargan de impartir clases en el área de computación, por lo que no pueden elaborar curso en Internet porque no saben diseñar páginas Web, además no existe una herramienta que les facilite crear sus clases y ponerlas en Internet.

PREGUNTA ¿Usaría una herramienta que le genere automáticamente su clase en Internet?

Objetivo

Conocer sí los docentes en general, utilizarían una herramienta que les genere automáticamente, cursos en Internet.

	Docentes en General	RESPUESTA
SÍ	86	92%
NO	7	8%
	93	100%



Hallazgo

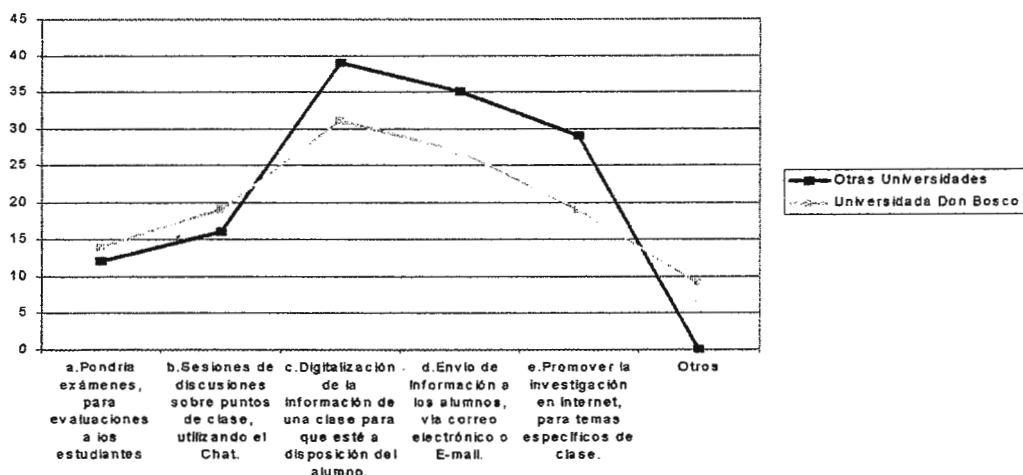
El 92% de los docentes utilizarían una herramienta que les facilite la creación de sus cursos académicos en el Internet, tanto para Docentes de la Universidad Don Bosco, como de otras Universidades, consideran que sería bueno implementar este tipo de proyectos ya que beneficiarían al estudiante grandemente, así como al docente.

Pregunta Marque con una "x" el uso que le daría a un asistente o herramienta para diseñar Usted mismo cursos o clases, que sean colocados en Internet

Objetivo

Detectar el uso que los docentes, darían a una herramienta que genere clases en Internet

	Otras Univ.	Porcentaje	UDB	Porcentaje
a. Pondría exámenes, para evaluaciones a los estudiantes	12	9%	14	12%
b. Sesiones de discusiones sobre puntos de clase, utilizando el Chat.	16	12%	19	16%
c. Digitalización de la información de una clase para que esté a disposición del alumno.	39	30%	31	26%
d. Envío de información a los alumnos, vía correo electrónico o E-mail.	35	27%	27	23%
e. Promover la investigación en Internet, para temas específicos de clase.	29	22%	19	16%
Otros	0	0%	9	8%
TOTALES	131	100%	119	100%



Hallazgo

Totalizando los porcentajes de las respuestas obtenidas en las respuestas de esta pregunta, reflejan el uso que los docentes le darían a una herramienta para crear clases en el Web, en la cual digitalizarían el contenido de su clase, así mismo muestra que utilizarían el Correo electrónico, para enviar cualquier tipo de información a sus alumnos, entre otros.

CONCLUSION

El desarrollo de esta encuesta permitió detectar que los docentes de la universidad Don Bosco, en primer lugar no conocen mucho sobre lo que es el diseño de Cursos en Línea, o documentos para el Web, algunos no saben que son Cursos en Línea, y tampoco conocen de una herramienta que les ayude a generar un Curso en Línea.

De igual forma, dicha encuesta permitió detectar que los docentes, de existir dicha herramienta la utilizarían para proporcionar el desarrollo de sus clases, introduciéndoles ilustraciones para hacerla menos aburrida, además establecer contacto con sus alumnos por medio de mensajes, para compartir información de tareas, ya sea de resolución o avisos relacionados con los temas que desarrolla en el aula. Así también permitió detectar que los docentes dentro de la Universidad Don Bosco, tienen derecho al uso del centro de computo, el cual permite que las personas puedan acceder a Internet, con lo cual se puede determinar que el acceso a Internet, no está restringido durante sus horas laborales,

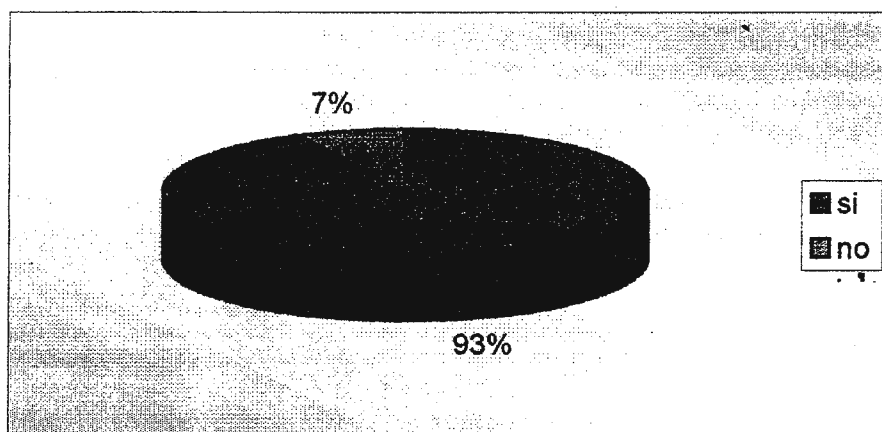
III.1.1.2 TABULACION DE ENCUESTAS ALUMNOS

PREGUNTA ¿Tiene acceso a Internet?

Objetivo

Determinar si los alumnos de la Universidad Don Bosco tiene acceso a Internet

	Frecuencia de Respuesta	Porcentaje
Si	139	93%
No	11	7%
TOTALES	150	100%



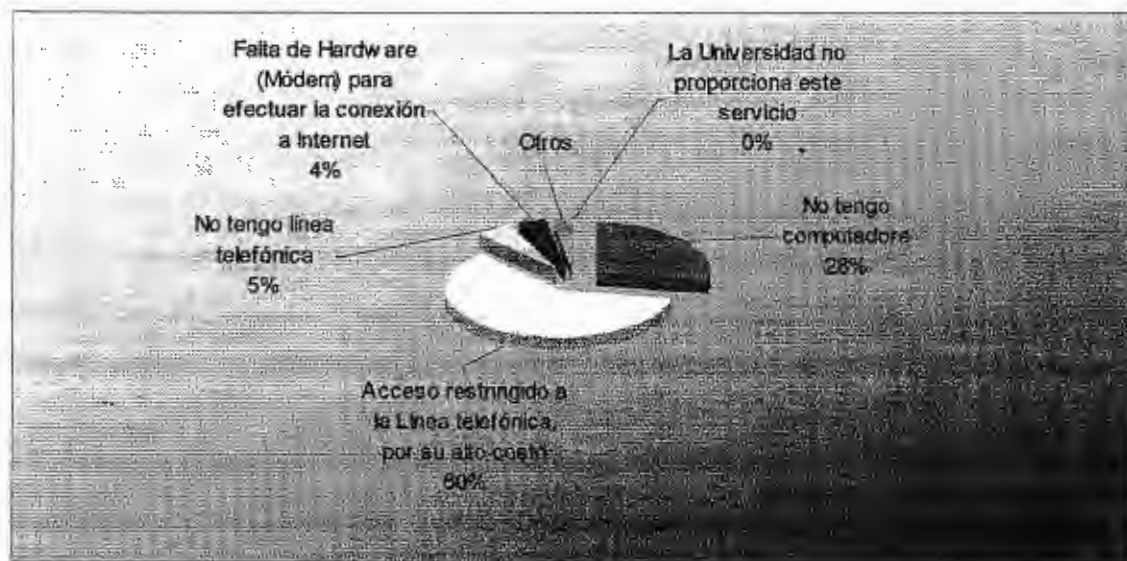
Hallazgo

La resolución de esta pregunta refleja que la mayoría de los estudiantes de la Universidad Don Bosco, tiene acceso a Internet, por lo cual se determina que podrían perfectamente revisar una clase puesta en Internet por el docente cuando él se los indique.

PREGUNTA ¿ Qué dificultades tiene para acceder a Internet?**Objetivo**

Determinar las dificultades que los alumnos tienen para acceder a Internet.

	Frecuencia de Respuesta	Porcentaje
La Universidad no proporciona este servicio	0	0%
No tengo computadora	42	28%
Acceso restringido a la Línea telefónica, por su alto costo	90	60%
No tengo línea telefónica	8	5%
Falta de Hardware (Módem) para efectuar la conexión a Internet	6	4%
Otros	4	3%
TOTALES	150	100%

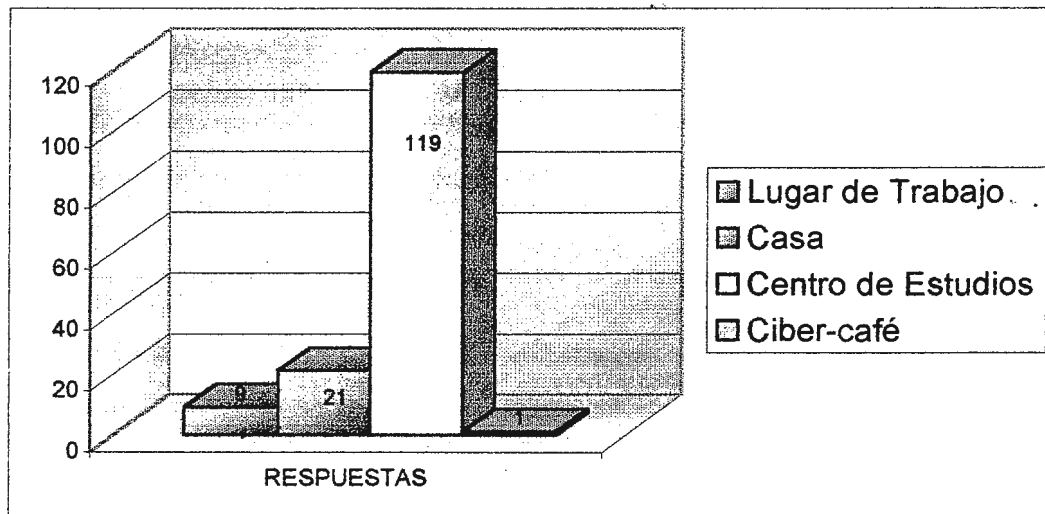
**Hallazgo**

Se determina que la mayoría de los alumnos, aunque tengan acceso a Internet, lo hacen de forma restringida, ya que el costo de la línea de teléfono, cuando lo hacen desde su casa, es bastante alto.

PREGUNTA ¿Desde qué lugar accesa a Internet?**Objetivo**

Determinar el lugar de acceso a Internet más conveniente para los alumnos de la Universidad Don Bosco.

	Frecuencia de Respuesta	Porcentaje
Lugar de Trabajo	9	6%
Casa	21	14%
Centro de Estudios	119	79%
Ciber-café	1	1%
TOTALES	150	100%

**Hallazgo**

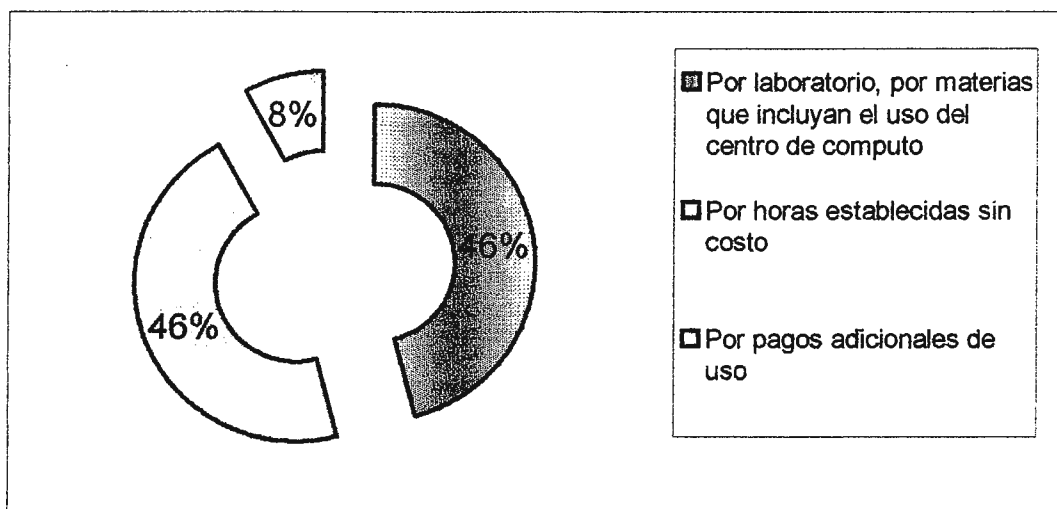
La mayoría de los alumnos de la Universidad Don Bosco, tienen acceso a Internet desde el centro de cómputo, ya que en su casa el costo de uso de la línea de teléfono es más costoso, y por ello prefieren hacerlo desde el centro de estudios.

PREGUNTA ¿Qué facilidad le proporciona su Universidad para acceder a Internet?

Objetivo

Conocer las facilidades de uso y de acceso a Internet, que tienen los Alumnos de la Universidad Don Bosco, proporcionados por la misma universidad.

	Frecuencia de Respuesta	Porcentaje
Por laboratorio, por materias que incluyan el uso del centro de computo	147	46%
Por horas establecidas sin costo	147	46%
Por pagos adicionales de uso	26	8%
TOTALES	320	100%



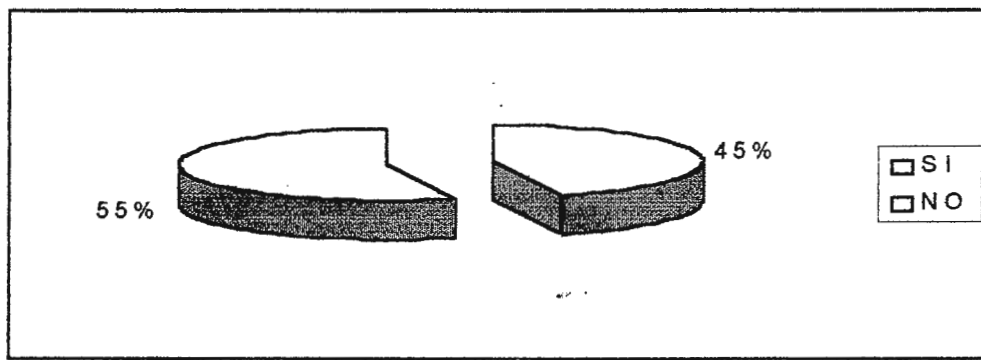
Hallazgo

Para este punto se determina que los alumnos tiene la facilidad de acceso a Internet, desde la Universidad, ya que se proporciona este servicio, por laboratorio, por horas para practicas, y en algunos casos por pagos adicionales, para los alumnos que no estudian ingeniería.

PREGUNTA ¿Sabe que son Cursos en Línea, o Cursos Online?**Objetivo**

Determinar el conocimiento que los alumnos tienen sobre lo que son Cursos en Línea

	Frecuencia de Respuesta	Porcentaje
SI	67	45%
NO	83	55%
TOTALES	150	100%

**Hallazgo**

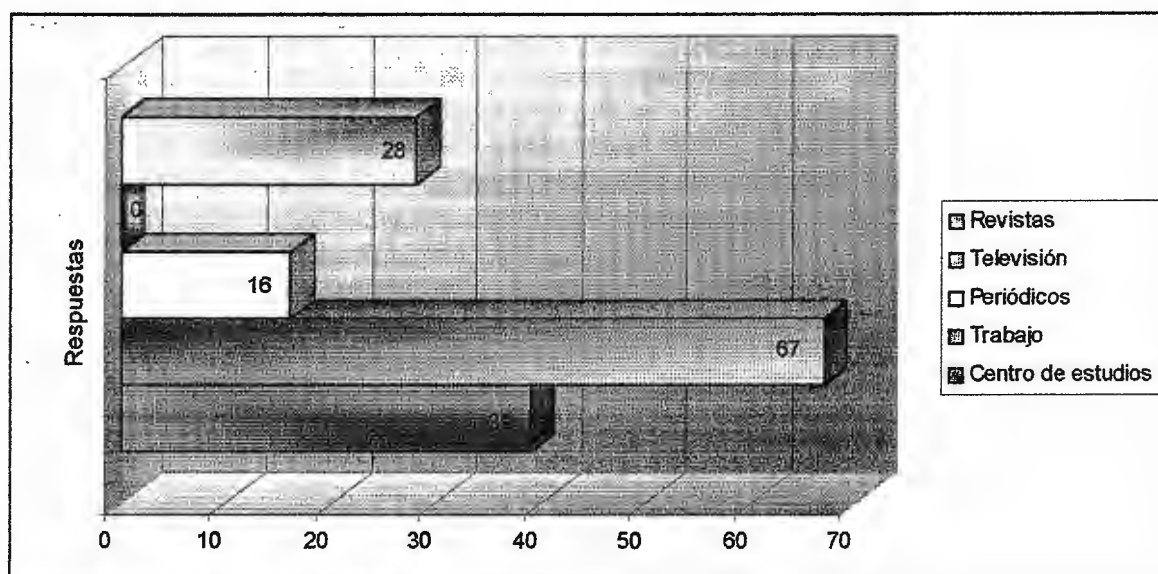
Se determina con este punto que el 55% de la población de alumnos encuestados en la Universidad Don Bosco, no conocen que son los Cursos en Línea, ya que no conocen ninguno y además en la Universidad no existe ningún servicio que proporcione una Clase en Internet.

PREGUNTA ¿Dónde ha escuchado sobre ellos?

Objetivo

Saber la fuente de información sobre los cursos en línea de los alumnos que han escuchado hablar de ellos.

	Frecuencia	Porcentaje
Centro de estudios	39	26%
Trabajo	67	45%
Periódicos	16	11%
Televisión	0	0%
Revistas	28	19%
TOTALES	150	100%



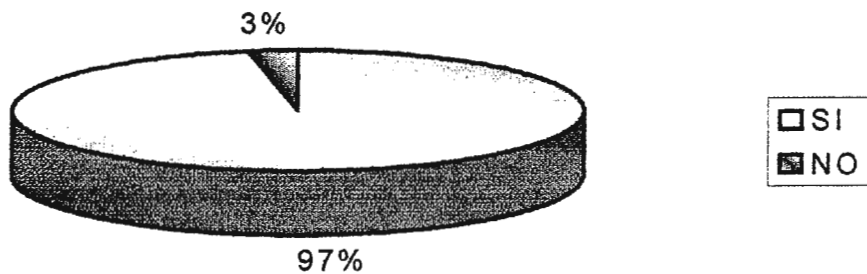
Hallazgo

Se determina que la mayoría de los alumnos que conocen sobre lo que son Cursos en Línea, han escuchado a cerca de ellos, en el lugar de trabajo, en el centro de estudios, y posteriormente en periódicos, ya que ahora en día hablar de Internet, es tan común y al mismo tiempo necesario, para mantenerse informado de todos los avances que van surgiendo constantemente.

PREGUNTA ¿ Le gustaría tener un Curso en Línea en su centro educativo?**Objetivo**

Saber si los alumnos de la Universidad Don Bosco están interesados en que se implemente un servicio de cursos en línea.

	Frecuencia	Porcentaje
SÍ	146	97%
NO	4	3%
TOTALES	150	100%

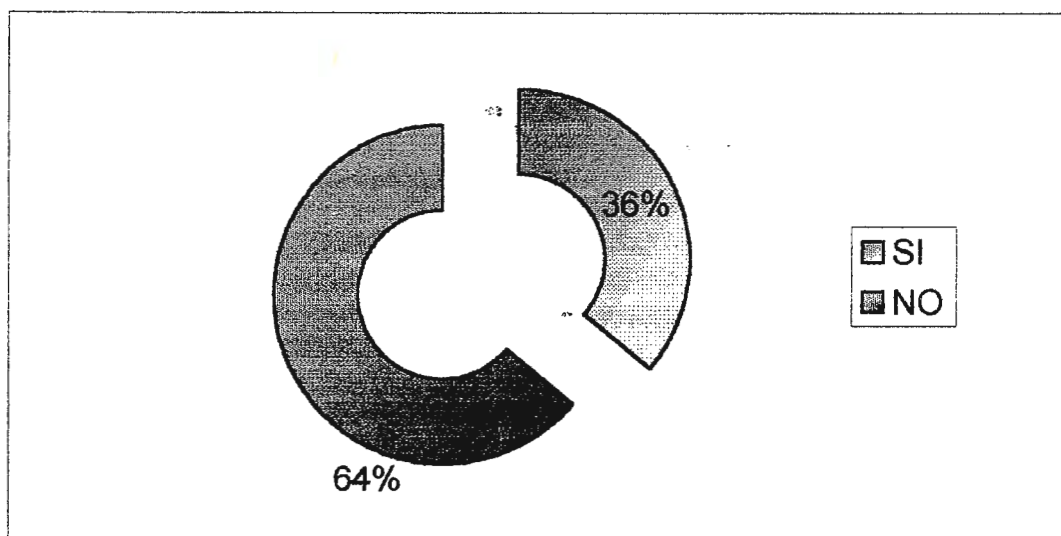
**Hallazgo**

Se determina que este proyecto urge de ser implementado, ya que el 97% de los alumnos encuestados están de acuerdo con que se desarrolle un Curso en Línea para la Universidad. Mientras que solo el 3% respondió que no, porque simplemente no necesitan ese tipo de metodología para aprender en su clase.

PREGUNTA ¿Ha visitado algún Curso en Línea (Curso Online)?**Objetivo**

Conocer si los alumnos que conocen sobre Cursos en Línea, han accedido alguno de estos cursos en línea.

	Frecuencia	Porcentaje
SÍ	54	36%
NO	96	64%
TOTALES	150	100%

**Hallazgo**

La mayoría de los alumnos que conocen sobre los Cursos en Línea han accedido a esos sitios, mientras que aunque algunos hayan escuchado y conozcan que son Cursos en Línea no han tenido la oportunidad de acceder a uno.

CONCLUSION

Con la elaboración de encuestas a una muestra de estudiantes de la Universidad Don Bosco, para detectar el conocimiento que tienen sobre lo que son los Cursos en Línea, la facilidad que tienen para acceder a Internet, y desde donde tiene el acceso, se lograron determinar que los factores importantes para el desarrollo de este proyecto, fueron que en la Universidad no existe un Curso en Línea que ayude a los alumnos a

estar más al tanto cuando perdieron una clase, actualizarse y repasarla para un examen, así como el acceso que se proporciona dentro de la Universidad para acceder a Internet, ya que muchos de los alumnos no lo hacen desde sus casas por el costo de la línea telefónica.

III.2 REQUERIMIENTOS DEL SOFTWARE

Hoy en día el software tiene un doble papel. Es un producto y, al mismo tiempo, el vehículo para hacer entrega de un producto.

¿Cómo seleccionar el Software adecuado?

Para saber que herramienta de programación utilizar, primero se establecen los requerimientos de los usuarios, lo que permitirá tomar las decisiones referente a que se utilizará para producir resultados satisfactorios, tomando como punto principal las características de cada herramienta de programación y las necesidades de los usuarios.

Las herramientas de Software utilizadas para el desarrollo del proyecto son las siguientes:

Linux

Como sistema operativo, el cual tiene la característica de que no necesita demasiado recursos en Hardware, esto significa que puede ser instalado en una maquina con procesador 386 como mínimo, y funciona de igual forma que en una Pentium III de 550 Mhz., esto es una ventaja ya que no se invierte recurso económico en ello, también facilita el código fuente, soporte en línea pues tiene una página en Internet para bajar información y actualizaciones de las versiones nuevas de dicho sistema operativo, estabilidad debido a la seguridad en el sistema operativo, proporciona el código fuente, para realizar cambios a dicho código según las necesidades del usuario, manipulación en el orden de ejecución de los procesos, ya que permite detenerlo en el momento en que de algún problema, posee un manejador de Bases de Datos que trabaja con sentencias SQL para manipular datos, fácil para MySql, compatibilidad con el lenguaje

de programación PHP3 el cual trabaja con sentencias HTML y JavaScript para la manipulación de datos, mediante páginas Web.

MySQL

Es un servidor de bases de datos SQL multi-usuario y multitarea. SQL es el lenguaje de bases de datos más popular en el mundo. MySQL consiste en una rutina (o "daemon") activada en el servidor (mysqld) y diferentes aplicaciones, basadas en programas y librerías, para el cliente de consulta.

Además trabaja con sentencias SQL para la manipulación de datos, puesto que es el lenguaje de bases de datos más popular. La versión de MySQL utilizada es la 3.22.32, aunque ya existen nuevas versiones, cabe mencionar que esta base de datos soporta el almacenamiento de archivos de imágenes, tipos de datos carácter y numéricos.

PHP

Es un lenguaje de programación que combina algunas características de Perl y de C para hacer interfaces WWW o programas CGI.

Para la Interface de consulta entre el servidor MySQL y el Web, se utilizó PHP3.0.15 Hypertext Preprocessor. PHP es un generador dinámico de páginas HTML que efectúa las consultas sobre MySQL, toma los resultados y elabora las tablas mostradas al usuario final. Todo esto se realiza dentro de una página que contiene los *tags* necesarios en HTML y las consultas dentro de un tipo de *tag* especial que es el que compila PHP. Estas páginas tienen terminación **.php3**.

Linux también trabaja con lenguajes de programación, la ventaja que ofrece este lenguaje es que los archivos hechos en código HTML, se pueden configurar en el servidor para que los almacene como archivos PHP, además es compatible con MySQL

Para la manipulación de datos, puede incluirse sentencias de JavaScript, que permiten procesar los objetos para generar páginas dinámicas que interactúen con la base de datos.

Incluye además el protocolo de comunicación TCP/IP, que permite la comunicación con otros servicios que usen protocolos de comunicación como el HTTP, IMAP, y POP3.

Apache

Servidor Web que corre bajo el sistema Operativo Linux

Para la instalacion y configuración de estos programas **ver Anexo 3** , el cual presenta los pasos, para dos tipos de instalacion, una en Linux y la otra en Windos.

III.3 REQUERIMIENTOS DEL HARDWARE

Para el desarrollo del proyecto se implementó, un laboratorio de prueba, instalando en una PC el Sistema operativo Linux y el Apache Server identificándolo como el Servidor, con objeto de formar una pequeña red; para que el Cliente, el cual es una PC con el sistema operativo Windows 98 accese mediante un Browser (Internet Explorer o Netscape) a la Interfaz, simulando así una conexión a Internet.

(Ver figura 2)

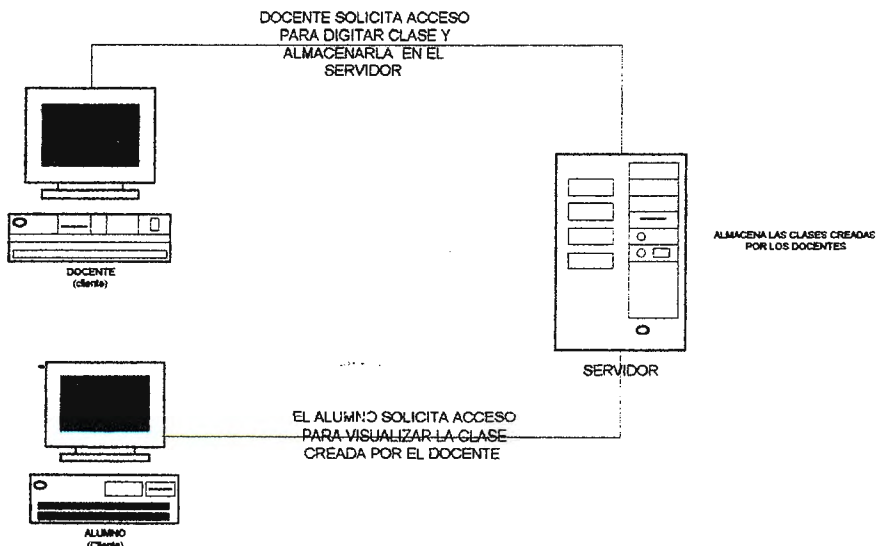


figura 2

El equipo de hardware utilizado, tiene las siguientes características:

Servidores

Dos PC

- Pentium 550 Mhz, con 128 MB de Memoria RAM, 10.8 GB de Disco Duro
- Pentium 533 Mhz, con 64 MB de Memoria RAM, 10.8 GB de Disco Duro

Clientes

Dos PC

- PC- 486, Velocidad de 100 Mhz. , 32 MB de Memoria RAM, 4Gb de Disco Duro, con Windows 98, Explorer 4.0, Office 97.
- PC- 486, Velocidad de 133 Mhz, 32 MB de Memoria RAM, 1 GB de Disco Duro, con Windows 98, Explorer 4.0, Office 97.

Además para la implementación de la red se utilizó el siguiente equipo:

Tarjetas de Red (4)

- NE 2000
- Sis 900 Fast Ethernet Adapter 10/100 bps
- Compact Netflex

Cable Crossover, para la conexión de red. (Ver Anexo 4)

La implementación de esta red tiene un costo, el cual se presenta en el **ANEXO 5**.

III.4 DIAGRAMAS DE FLUJO DE DATOS (DFD)

III.4.1 DEFINICION Y NOTACION UTILIZADA EN UN DFD

El diagrama de flujo de datos (DFD) es una técnica que representa el flujo de información y las transformaciones que se aplican a los datos desde la entrada hasta la salida.

La transformación de datos de entrada en salida por medio de procesos puede describirse en forma lógica e independiente de los componentes físicos asociados al

sistema. El análisis de flujo de datos permite determinar y estudiar las áreas de interés al examinar los datos que entran en el proceso.

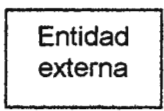


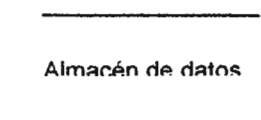
Ahora podemos preguntarnos *¿Qué es el análisis de Flujo de Datos?*

El análisis de flujo de datos estudia el empleo de los datos en cada actividad. Documenta los hallazgos con diagramas de flujo de datos que muestran en forma gráfica la relación entre procesos y datos, y en los diccionarios de datos que describen de manera formal los datos del sistema y los sitios donde son utilizados.

El análisis de flujo de datos utiliza el Diagrama de Flujo de Datos, el cual es una herramienta gráfica que describe y analiza el movimiento de datos a través de un sistema, ya sea que este fuera manual o automatizado, incluyendo procesos, lugares para almacenar datos y retrasos en el sistema.

El método para el análisis de flujo de datos seleccionado para el presente trabajo es el de Yourdon Inc, una compañía de consultoría y desarrollo profesional, y la simbología utilizada es la siguiente:

Notación DFD básica

	<p>Un productor o consumidor de información que reside fuera de los límites del sistema a ser modelado.</p>
	<p>Un transformador de información (una función) que reside dentro de los límites del sistema a ser modelado.</p>
	<p>Un flujo de datos; la cabeza de la flecha indica la dirección del flujo de datos.</p>
	<p>Un depósito de datos que se va a almacenar para uso por uno o varios procesos; puede ser tan simple como una memoria intermedia o cola, o tan sofisticado como una base de datos relacional.</p>

La ilustración del flujo de datos del sistema de la Interfaz se presenta en **Anexo 6**

III.4.2 DESCRIPCION DE LOS DIAGRAMAS DE FLUJO

En la realización de este punto, se utiliza la *Especificación del proceso (EP)*⁹ para describir todos los procesos del modelo de flujo de datos en el nivel final del diagrama. El contenido de la especificación del procesamiento puede incluir una narrativa, una descripción en *lenguaje de descripción de programa (LPD)* de algoritmo de proceso, ecuaciones matemáticas, tablas, diagramas o gráficos. Al proporcionar un EP que acompañe cada burbuja del modelo de flujo, se crea una mini-especificación que sirve como primer paso para la creación de la especificación de requisitos de software y constituye una guía para el diseño del componente del programa que implementará el proceso.

Para describir los procesos empleados en el funcionamiento de la interfaz, se utiliza una EP Narrativa, la cual se detalla en el **Anexo 7**.

⁹ Ingeniería del software, un enfoque práctico, Roger Pressman, Pág. 220, 221

CAPITULO IV

ELEMENTOS PRINCIPALES DE LA INTERFAZ

IV. 1 ELEMENTOS PRINCIPALES

Una breve descripción o una explicación corta de lo que hacen los elementos principales de un sistema cualquiera, siempre beneficia a los usuarios, ya que en algunos casos muchos de ellos, conocen poco sobre computación; por ende siempre buscan una guía que les brinde la información necesaria, de forma sencilla sin que se usen muchas palabras técnicas. Es por ello que para los docentes involucrados en el uso de ésta herramienta de apoyo a la enseñanza superior, específicamente de la Universidad Don Bosco, tengan un amplio conocimiento de lo que es cada uno de los elementos que forman parte de la Interfaz automática para elaborar clases en línea, se presenta en este capítulo un resumen de los principales elementos de dicha herramienta, como lo son:

El editor, donde el docente digita el las clases o la explicación del contenido teórico que impartirá a sus alumnos, al mismo tiempo podrá incluir **objetos** como: imágenes, sonidos, graficas, applets de Java, laboratorios virtuales, entre algunos, los cuales le ayudaran a ilustrar mejor su clase. De igual forma podrá desarrollar lo que son exámenes teóricos que por el momento tienen nada más la función de un **quiz de ejercicio** sin ponderación o valor real en las notas de los estudiantes. También el docente mantendrá una comunicación con sus alumnos mediante los **comentarios** los cuales son enviados por ellos mismos; estos mensajes capturan el nombre del usuario alumno, para que el profesor sepa que estudiante esta enviando ese mensaje.

Es importante mencionar que estos elementos están incluidos en la página principal del docente ya que es él quien genera o crea las clases en línea. Existe también la **página para el alumno**, y al igual que los docentes estos deben registrar sus datos llenando un formulario para ello y esperar a que el administrador de la base le de de alta, la página principal de alumnos contiene los elementos de **clases**, que es donde ellos visualizan las clases creadas por sus docentes; **comentarios**, la cual les permite enviar mensajes a sus maestros, **evaluación** donde el estudiante resuelve un

cuestionario teórico, de la clase que el docente le indica, así el alumno puede medir el grado de conocimiento que adquirió al final de una clase.

La aplicación presenta una página principal (Ver figura 3), con dos opciones, una para los docentes y otra para los alumnos, el docente al hacer clic en Docente iniciará una sesión que le permitirá crear sus clases en línea y el estudiante al hacer clic en Alumno visualizará las clases que el docente ha puesto a su disposición, de no tener un usuario y una contraseña para el acceso en ambos casos, deben registrarse, haciendo clic en la palabra "**REGISTRASE**" la cual presenta otra página que contiene un formulario, en deberán digitar sus datos personales; luego de llenar este formulario, se hace clic en *Enviar* y se espera un tiempo prudencial a que el administrador de la Base de Datos, apruebe la solicitud de acceso, luego de esta espera, visitar nuevamente el sitio y escribir el usuario y contraseña seleccionados.

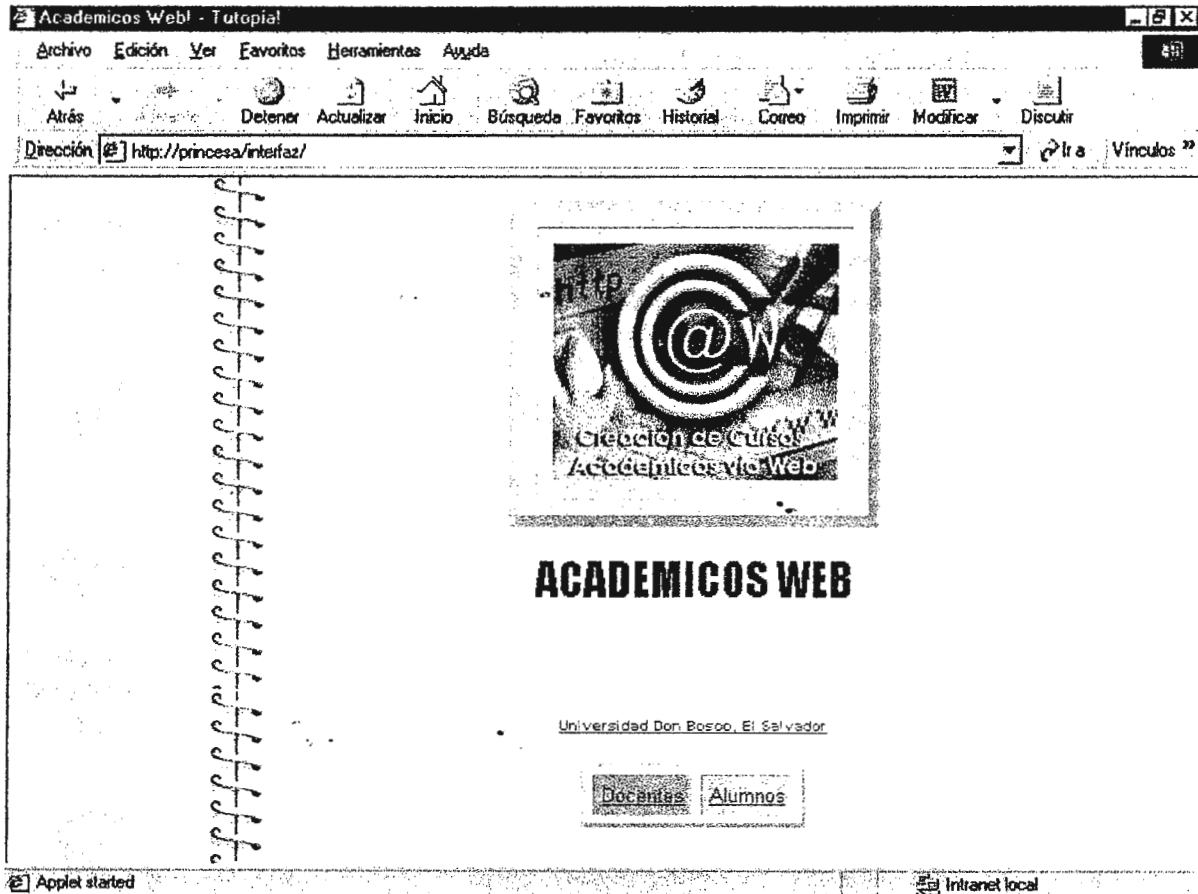
Ahora el docente al digitar el usuario y la contraseña, llega a la pantalla (ver figura 4), la cual presenta las opciones de crear tema, subtema, clase, objetos, evaluación, mensaje. En la opción de Objetos, es donde el docente subirá los archivos como imágenes, gráficos, sonidos, realidad virtual, etc. Que ayudaran a ser más ilustrativa la clase. (ver figura 5), luego de subir los objetos, se definen los objetivos de la clase, luego los temas y subtemas, posteriormente a ello, se puede hacer uso del Editor de la Interfaz donde se digitara la teoría de cada uno de los subtemas especificados, el editor ayudará al docente a crear la página Web de la clase, el cual coloca de forma automática los comandos de la codificación HTML, para ser vistas en el Browser, así mismo permite modificar las propiedades de texto, incluir tablas, imágenes, u otros objetos seleccionados por el docente para incluir en su clase y hacerla más ilustrativa.

Posteriormente el docente puede incluir las tareas o ejercicios para medir la capacidad de entendimiento de los alumnos desde la opción de "evaluación-ejercicios", siempre en el menú y además tendrá comunicación con sus alumnos vía correo electrónico desde la opción "Leer comentarios".

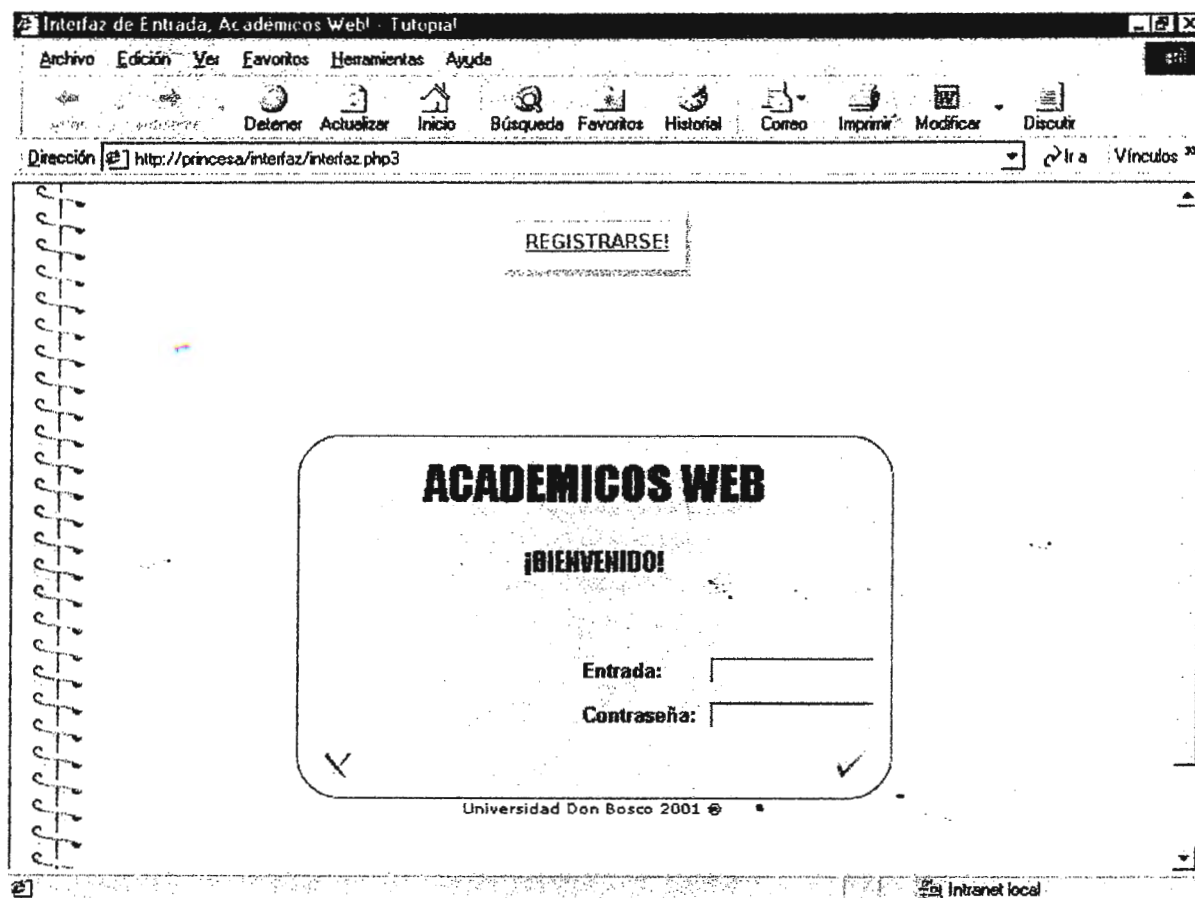
De igual forma el alumno podrá tener acceso a las clases de su maestro, desde la opción "Alumnos", donde seleccionará "Consultar clase" y luego digitará el nombre del maestro y le presentara las diferentes clases para ese docente; así mismo puede

enviarle comentarios al docente vía E-mail, y resolver preguntas en las evaluaciones teóricas incluidas al final de cada clase.

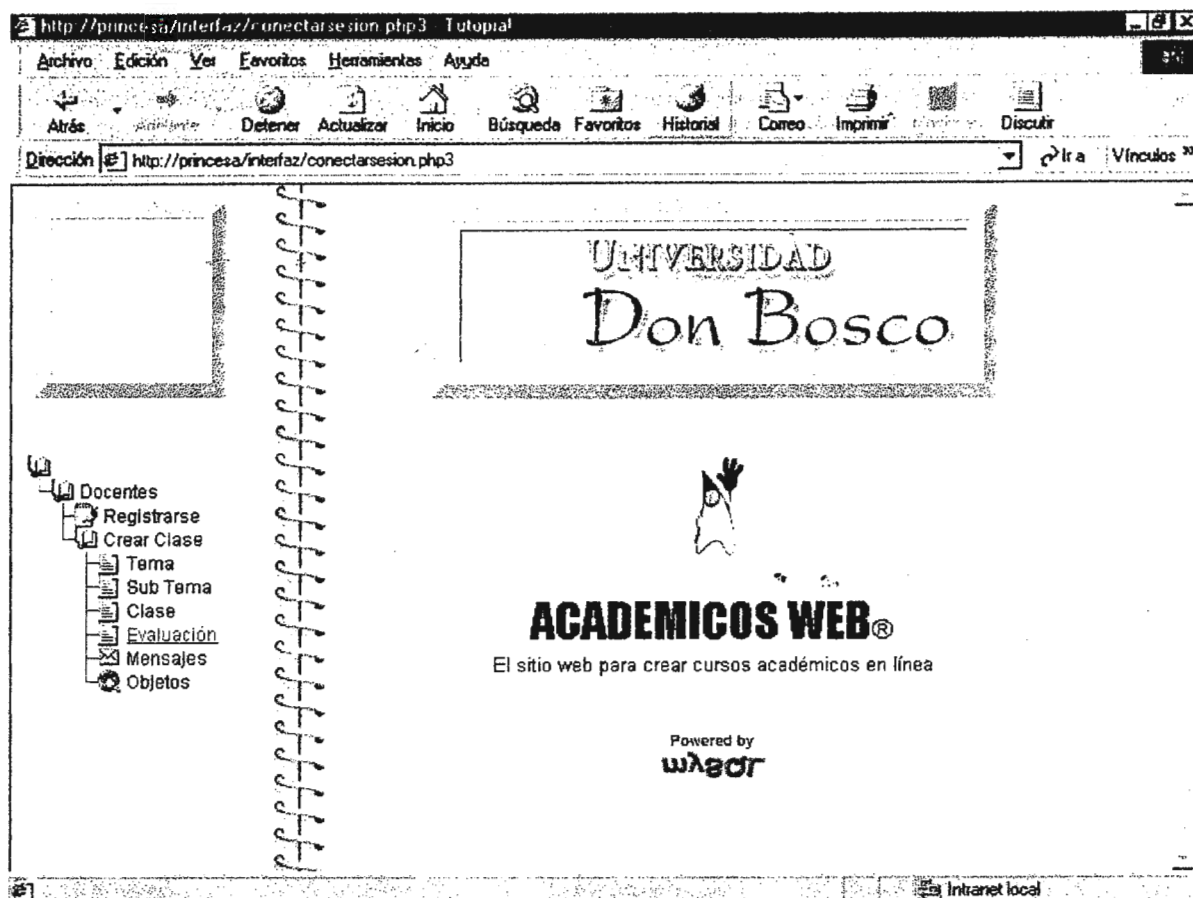
Y para tener un mayor conocimiento del funcionamiento de cada uno de los elementos, puede referirse al manual del usuario incluido en los anexos.



(Figura3)



(Figura 4)



(Figura 5)

Y para conocer el funcionamiento de los elementos principales de la Interfaz automática, se detallan de forma general a continuación:

IV.2 EL EDITOR

Básicamente funciona de la siguiente manera:

En el se digita la información teórica de la clase que el docente imparte, la ventaja de este editor es, la facilidad que proporciona al colocar el código HTML que permite generar las páginas Web, automáticamente, de esa forma, los docentes que no conocen de la programación en HTML solo necesitan digitar sus clases e incorporar características especiales en los temas etc., incluir además direcciones de otros sitios en Internet como referencia de la clase, incorporar objetos como graficas, dibujos, gif

animados, entre otros para ilustrar sus clases, también caracteres especiales, los cuales son seleccionados de una tabla incorporada en el editor. (Ver figura 6)

Luego de esto el docente genera la clase, la cual está lista para colocarla en el Servidor, también el editor contiene un botón que permite ver las clases creadas para verificar como los alumnos la verán, abriendo en el Browser la clase ya terminada.

Es importante mencionar que este editor, posee muchas de las características de Word de Office, Notepad o Block de notas de Windows o el Edit de MS DOS, la diferencia está en que este editor genera un documento con etiquetas HTML, que es el código de programación utilizado para generar páginas en Internet.

The screenshot shows a web browser window with the address bar displaying a URL. The main content area is titled "Generar Clase" and contains several form fields and sections:

- ENCABEZADOS**: A section with buttons for G, M, and P.
- ESTILOS**: A section with buttons for N, C, and S.
- ESPECIAL**: A section with buttons for F, P, and L.
- JUSTIFICADO**: A section with buttons for I, C, and B.
- LETRA**: A section with a text input field containing "Arial", a "Negro" button, and a "2" button.
- LISTA**: A section with a "LISTA" button.
- Título Clase:**: A text input field.
- Dirección e-Mail:**: A text input field.
- SITIOS INTERESANTES**: A section with two columns of text input fields labeled "Dirección(URL):" and "Nombre Dirección:".
- CUERPO DE LA CLASE**: A section with multiple text input fields for the main content of the class.

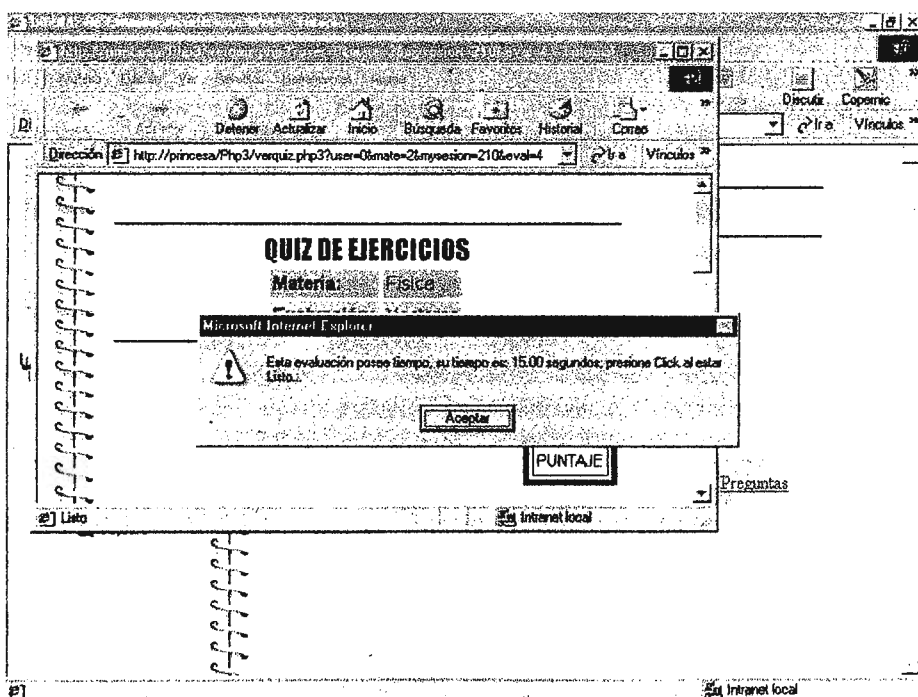
(Figura 6)

IV.3 EL QUIZ DE EJERCICIOS

Funciona de la siguiente manera :

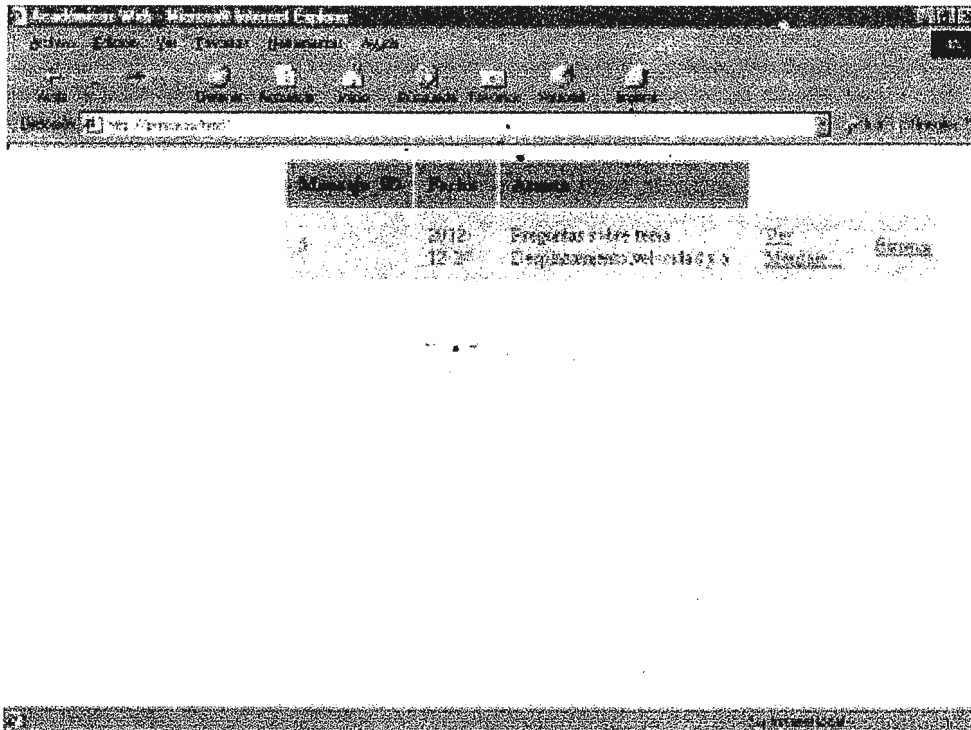
Consiste en que el docente vaya formulando su test de evaluación, procediendo de la siguiente manera, primero el docente selecciona la materia, una vez seleccionada la materia, hacer clic en "Hacer evaluación" (ver figura 7), este envía a otra pantalla en la que se define si la evaluación tendrá tiempo de duración para ser resuelto por los alumnos, si es así el docente especifica la cantidad de tiempo en segundos, luego de esto se hace clic en "Preguntas" enviando a otra pantalla, donde se digitan las preguntas, y una vez elaborada la pregunta se hace clic en Añadir, luego por cada pregunta que se formule en la evaluación se introducirán las respuestas, al hacer clic en la opción respuestas, envía a otra pantalla donde se digitan las respuestas para dicha pregunta, este paso posee una característica, el cual consiste en un casilla chequeada que indica que esa es la repuesta correcta, teniendo cuidado de quitar el cheque cuando la respuesta no sea la correcta, por que de lo contrario, enviará un mensaje de error indicando que ya existe una respuesta correcta.

Una vez terminadas las preguntas y sus respuestas para la evaluación, el docente puede visualizar su hoja de ejercicio antes de que la vean los alumnos.



IV.4 LEER COMENTARIOS

El docente solamente visualiza los comentarios enviados por sus alumnos, funcionando como el correo electrónico, el cual le permite además seleccionar el mensaje que quiera ver en ese momento, ya que estos mensajes almacenan la fecha, quien lo envía y sobre que es la pregunta. (Ver figura 8)



(Figura 8)

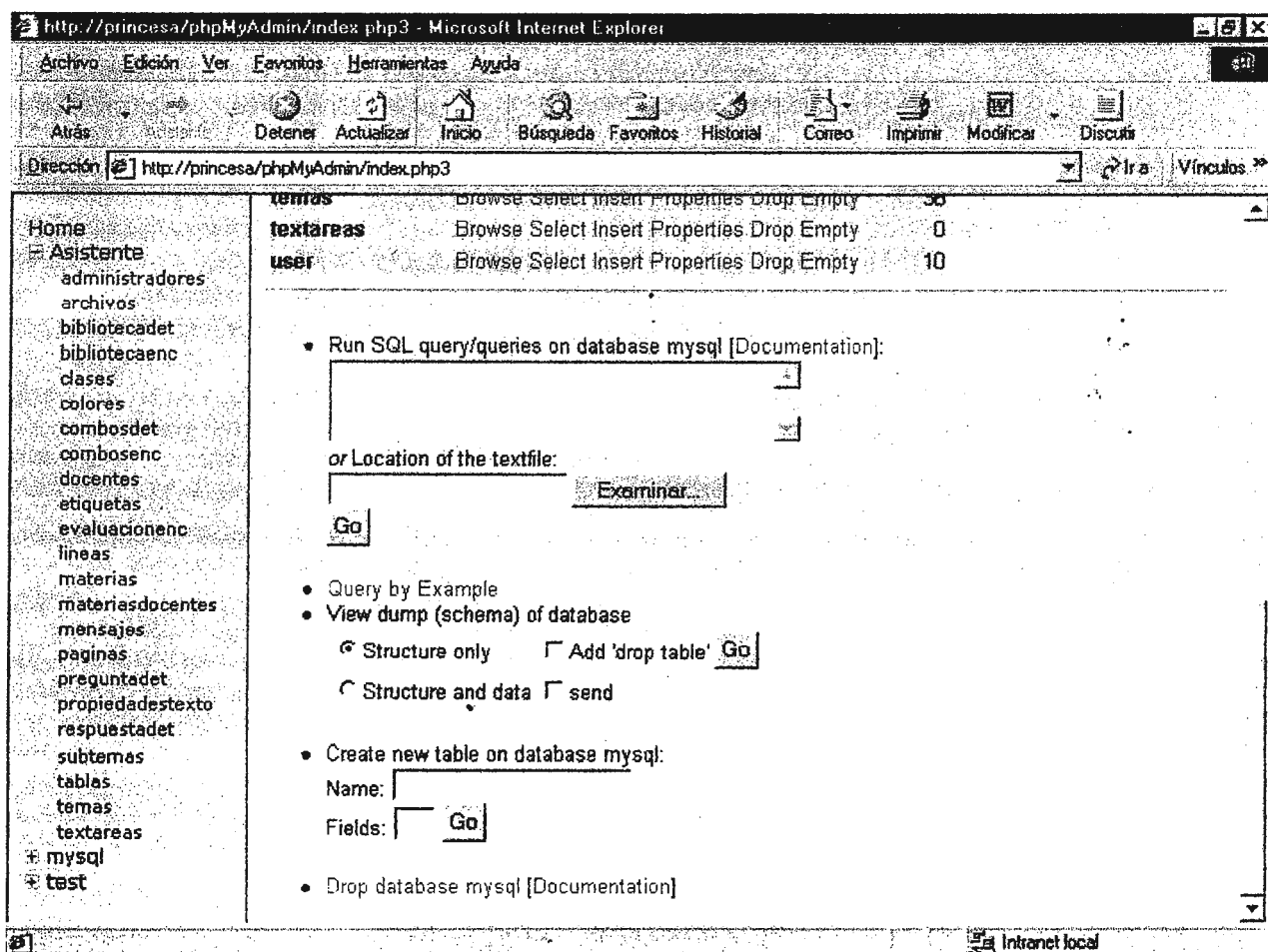
IV.5 PHPMyAdmin

Para la administración de la base de datos se presenta la herramienta PHPMYADMIN, que funciona con el objeto de facilitar al administrador de la base la solución de problemas relacionados con programas, las tablas y los campos o registros de cada tabla, los tipos de datos que maneja la Base de Datos. (Ver figura 9).

El administrador de la base para asignar derechos de los usuarios debe de ingresar a la tabla User

Para la utilización de esta herramienta el cual es un programa que permite al Administrador de la Base de Datos, controlar la información que se maneja en la base de datos, la asignación de permisos y derechos para los usuarios que ingresan a la aplicación, así mismo permite cambiar campos de las tablas de la Base de Datos, hacer copias de respaldo de la base completa con toda su estructura o solamente los datos que esta contiene; también permite generar nuevas tablas con sus respectivos campos y las características para cada uno; y realizar cambios en los programas desarrollados en PHP, y probar los cambios efectuados.

Para tener mayor referencia, se presenta el manual técnico en los anexos, donde se explica con mayor detalle el funcionamiento de este programa.



(Figura 9)

CAPITULO V

HERRAMIENTAS QUE PROPORCIONAN AYUDA PARA EL DESARROLLO DE SITIOS WEB

V.1 INTRODUCCION

Internet permite encontrar diferentes sitios orientados a las necesidades de los usuarios, ya sean estas comerciales, educativas, gubernamentales, de entretenimiento, modas, social, religioso, culinario, científico, etc.

Pero para este proyecto nos interesa conocer que otras herramientas existen en Internet, que permitan la elaboración de sitios Web, específicamente hablando en el área de la Educación. En tal sentido existen algunos sitios que proporcionan manuales, para aquellas personas que conocen muy poco sobre la elaboración de páginas Web y lo concerniente al diseño y/o colocación de documentos en Internet, con el propósito de orientar a las personas que necesitan elaborar documentos sin mayor problema; Muchos de estos manuales, ilustran muy detalladamente los pasos que se seguirán durante la creación de páginas Web, aunque diseñar con sencillez una página web es poco frecuente, ya que las imágenes y los recursos tecnológicos son una tentación en la que muchos suelen caer. Por lo que el concepto de Interactividad está muy deformado y suele pensarse que esta palabra se refiere a lo que se conoce como multimedia, cuando Multimedia es la capacidad que posee la Internet de integrar medios (imagen, sonido, cinética); de hecho este concepto se relaciona mucho con la enseñanza, ya que las ilustraciones son muy importantes en algunas materias.

De hecho existen muchos sitios que permiten crear páginas web, que son muy distintos a los sitios, una página web es un espacio donde se coloca toda clase de información pero que en realidad no es muy compleja, mientras que un sitio Web es colección de páginas que van conectadas entre si, manteniendo una relación mediante los vínculos, estos además de proporcionar información, muchos de ellos interactúan con datos, los cuales en la mayoría de los casos son extraídos desde una base de datos, almacenada en un servidor; esa es la diferencia entre una página web y un sitio web. También es importante mencionar la existencia de lugares que además de crear páginas Web

permite almacenarlas dentro de su servidor, estableciendo un espacio gratuito que generalmente es de 2 Mb, por lo que el uso de muchas fotografías, gráficos o dibujos, se ven limitados puesto que muchas veces consumen demasiado espacio en memoria y la presentación de las páginas en el Browser ¹⁰ se vuelve muy lenta.

Por otra parte para el diseño de páginas Web, existen diferentes herramientas de las cuales podemos mencionar el FrontPage, Visual Page, Homesite, Geocities, Angelfire (de los más reconocidos), que permiten el diseño y creación de las mismas, ayudando mucho a aquellas personas que conocen muy poco sobre HTML (Hiper Text Markup Lenguaje), el lenguaje de programación de Páginas Web, de hecho también existen en Internet muchos sitios que proporcionan manuales o tutores con ayuda: como formularios, donde los usuarios digitan la información que necesitan en su página, luego de esto, el tutor o ayudante indica al usuario que debe de grabar en el servidor las imágenes o fotos que desee incorporar en su página web, brindando así mayor facilidad para comprender y conocer más sobre el diseño de aplicaciones para el Web; tomando en cuenta que es solamente la creación de Páginas Web. Así mismo estas páginas pueden ser colocadas en otros sitios como Geocities, Gratis Web, Angelfire, etc, que permiten almacenarlas sin costo por servicio de creación y/o alojamiento, aunque como se mencionó anteriormente muchos de estos sitios tienen asignado el espacio justo en el servidor.

También existen herramientas que permiten la generación de sitios Web, los cuales interactúan con una base de datos mediante formularios para capturar la información que la base de datos necesite, o para lo que este diseñado el sitio.

A continuación se presenta una breve descripción de algunos sitios que se consideran como herramientas bastante aceptables y fáciles de usar, y que su funcionamiento es muy parecido al de esta Interfaz, así se tiene: Publisher de MsOffice versión PYME, Nuestro.elsalvador.com del Diario de Hoy. Aunque se sabe que existen otros sitios que permiten diseñar o crear sitios de forma automatizada, pero que en realidad no tienen

¹⁰ Es un programa (como Nestcape, Mosaic, Microsoft Internet Explorer) que permite ver la información en el W.W.W. en un formato gráfico.

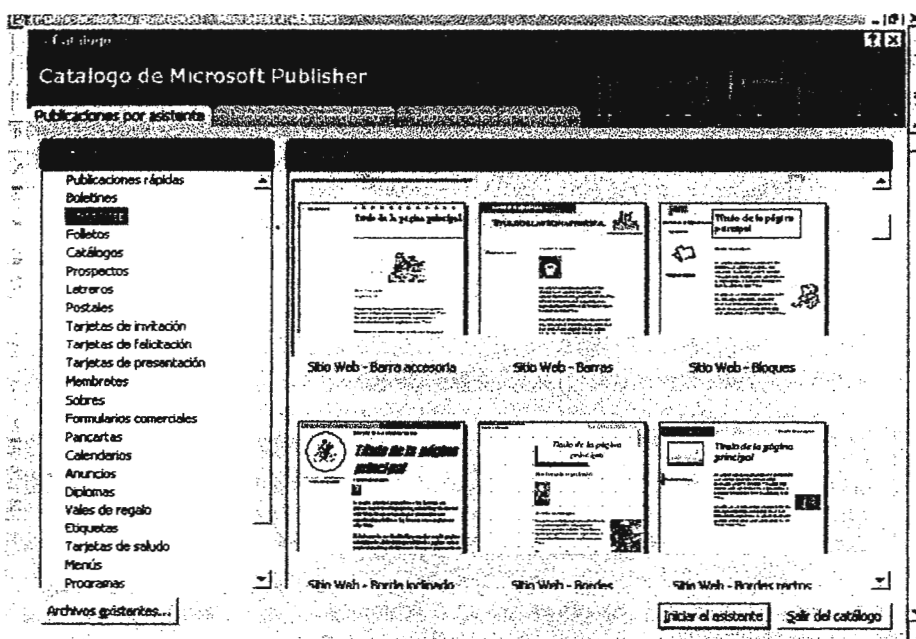
mucha funcionalidad, para el objetivo que se persigue con esta aplicación, sitios para desarrollar Cursos o Clases en Internet.

Para que se tenga una idea de cómo funciona estas otras herramientas, se describen de forma general a continuación :

V.2 Publisher de MsOffice

Permite crear un sitio Web, el cual contiene un tutor que guía al usuario durante toda la generación del sitio, este tutor consiste en que el usuario escoja los elementos que desea o necesita incluir en su sitio, así al iniciar selecciona de una gama de estilos de Hojas Web la cual será la principal, luego permite cambiar las combinaciones de colores (las cuales tienen un nombre) de la hoja seleccionada, luego pregunta que páginas se desean incluir en el sitio, los cuales pueden ser: formularios, tablas y/o gráficos, y para cada página adicional crea una nueva, las cuales son editadas al final, luego de que ha seleccionado todo lo necesario para el sitio Web, se presiona el botón "Finalizar" y se procede a la edición de las Hojas, desde la principal en la se cual genera un índice, hasta las adicionales, digitando en ellas la información que se desea publicar.

Luego de esto se tiene listo el sitio Web, para ser depositado en un servidor Web, ya sea pagado o gratuito como los mencionados anteriormente. (Ver figura 9)



(Figura 9)

V.3 Nuestro.elsalvador.com

Esta es una aplicación creada en el exterior, que básicamente funciona igual que el Publisher, el cual proporciona las herramientas a todos los usuarios, para que puedan elaborar sus propias páginas electrónicas completamente gratis. Este proyecto contiene 16 canales de tópicos como deportes, educación, fútbol, gente, cultura, religión, negocios, empresas, noticias, y muchas cosas más. Es bueno que el canal de educación haya sido dividido en diferentes niveles puesto que se pueden encontrar desde tesis, currículos, y temas de materias. Para la elaboración de páginas Web, existen 20 formatos de hojas, de las cuales las personas seleccionan de su agrado, pero si alguna persona puede diseñar su propio sitio, puede hacerlo sin ningún problema.

Pero para elaborar las hojas electrónicas no se necesita conocer de programación HTML y el FTP (servicio para colocar una página en Internet), pues esta aplicación lo permite, de una forma tan sencilla pues la personalización del sitio se hace mediante un formulario que el usuario va llenando con sus datos personales, luego de llenar este

formulario y categorizar el sitio, da clic en el botón "Create Site" y posterior mente lo envía a seleccionar el estilo de hoja web. (Ver figura10)

Create a Web Site on El Diario de Hoy
Login or Set Up a New User Profile

New Users:

[Click here to set up a new user profile](#)

Registered Members - Log in here!

[Get help with logging in.](#)

Username:

Password:

[Lost password? Click here!](#)

[Cancel](#)

[Privacy Policy](#)

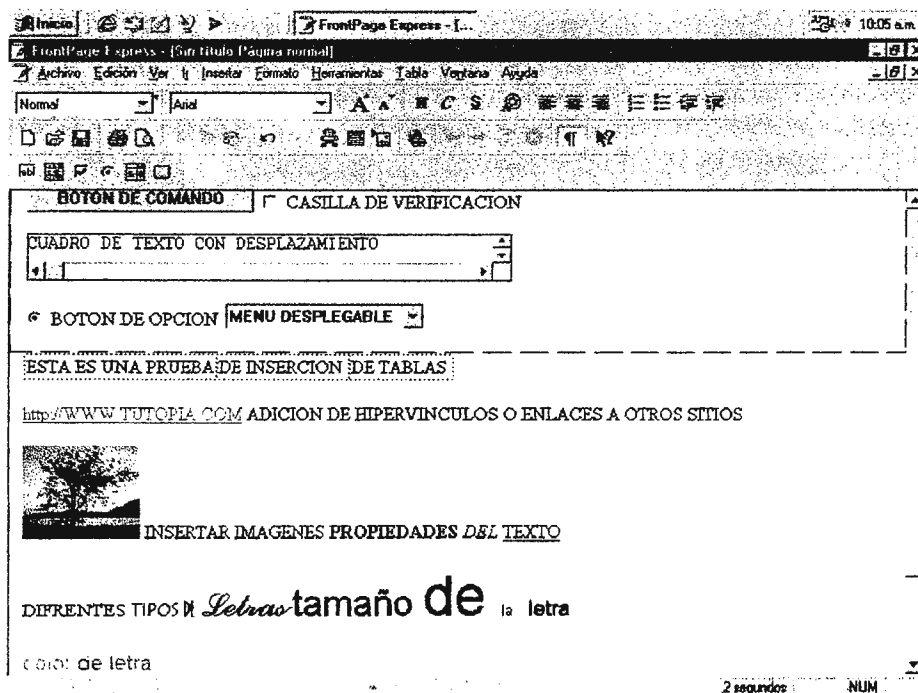
KOZ A KOZ.com
Community

(Figura 10)

V.4 FrontPage

Es una herramienta de desarrollo, que forma parte de Windows 98, la cual permite diseñar sitios Web elaborando páginas Web, proporcionando las opciones necesarias para crear formularios que interactúen con la base de datos, para capturar la información correspondiente.

FrontPage posee una barra de herramientas, formada por las opciones de tipo de letra o fuente, tamaño, propiedades como negrita, cursiva, color, etc, alineaciones o justificaciones de texto, adición de tablas, objetos como casilla de texto, verificación, de opción, botón de comando, menú desplegables, entre otros, así mismo permite la inserción de imágenes, realizar vínculos, incorporar sonidos, cambiar fondos, adicionar listas enumeradas o con viñetas de caracteres; esto aparte de los comandos normales de adición, eliminación, modificación, guardar, abrir, nuevo, imprimir, vista preliminar, etc. También permite ver el código HTML, que va generando automáticamente, el que genera las páginas Web. (Ver figura 11)



(figura 11)

A continuación direcciones de Internet, de las cuales puede obtenerse diferentes script con ejercicios de física, las cuales los docentes pueden utilizar para sus clases.

http://www.cyberworldcorp.com/corpsite/products/products_downloads.html

<http://fisica.ciens.ucv.ve/~felix/JavaScript/zeros.html>

<http://fisica.ciens.ucv.ve/~felix/JavaScript/zeros.html>

CONCLUSIONES

Académicos Web, ya como un prototipo, está en la capacidad de ofrecer los siguientes servicios:

- Puede manejar seguridad por alumno o por docente, y no permite que alumnos puedan ingresar a la consola de docentes para crear las clases o evaluaciones.
- El sistema es capaz de manejar por ende dos consolas, una para el alumno y otra para el docente.
- Proporciona la alternativa de que las clases no solo puedan quedar plasmadas en papel, sino también en discos flexibles.
- Mantiene un historial de todas las clases que un docente crea en la herramienta, conservando la riqueza de conocimientos que ellos nos proporcionan.
- Tiene la capacidad de incorporar a cada clase diferentes objetos, como: imágenes (*.GIF o .JPG), videos (*.MPG), sonidos (*.WAV), archivos de realidad virtual (VRML *.WRL) y Applets ó aplicaciones en JAVA (*.CLASS) que proporcionan una forma de agregarle interactividad a los cursos académicos. Para esto, el sistema consta de una consola en la cual puede subir archivos de estos tipos (DESCARGA DE ARCHIVOS) y ser almacenados en el servidor, bajo el nombre de usuario del docente, esto funciona de tal forma que el docente solo puede utilizar los objetos que él ha descargado en el sistema.

enseñanza

- La herramienta proporciona facilidad para la creación de cursos y evaluaciones, su objetivo es llevar de la mano al docente en la creación de los mismos, a su vez proporciona a los alumnos un fácil acceso a las clases y evaluaciones.
- *Académicos Web* posee un editor creado con los lenguajes HTML, PHP3 y JAVASCRIPT el cual facilita la creación de las páginas que puede contener una clase elaborada por el docente. El editor es capaz de funcionar fuera del Internet mientras el docente crea la página de la clase, a su vez proporciona funciones para facilitar la creación del curso, como por ejemplo:
 - Tipos de encabezado.
 - Tipos de letra.
 - Tablas de símbolos especiales.
 - Función de ayuda.
 - Separadores de líneas (entre una línea de texto y otra) y párrafos.
 - Creador de Vínculos (Links) automatizados.
 - Creador de líneas con diferentes altos y anchos.
 - Creador de subíndices y superíndices.
 - Controles de justificado (Izquierda, Centro y Derecha).
 - Creador automático de tablas, en los cuales se pueden definir el número de filas y columnas así como el color de las celdas y bordes del mismo.
 - Permite manejar diferentes tipos de letras para poder aplicar en un determinado texto.
 - Es capaz de insertar objetos en la clase como: imágenes, videos, sonidos, archivos de realidad virtual y applets interactivos.
 - Es capaz de recobrar la información de una página de la clase en la misma plantilla del editor con objeto de modificarla para sus respectivas actualizaciones.
 - El editor permite la inserción de una clase previamente hecha en cualquier otro editor de páginas web, facilitando así la generación de la clase.

- El sistema proporciona un módulo especial para la *creación de evaluaciones o quiz de ejercicios*, el cual es capaz de mostrar una evaluación que posea múltiples preguntas y cada pregunta puede poseer múltiples respuestas; también es capaz de calcular el puntaje al final de la evaluación y poseer un tiempo indefinido de duración o, puede desarrollarse en un tiempo estipulado por el docente (la unidad de tiempo es el segundo).

La evaluación está clasificada por docente y materia y puede repetirse cuantas veces desee el alumno.

- La herramienta consta de un módulo para la *administración de mensajes*, el cual facilita la comunicación entre el docente y el alumno. Permite al alumno emitir un mensaje hacia un docente determinado y éste automáticamente toma los datos del alumno que lo envía. De esta forma el docente puede enterarse de la procedencia del mismo.
- Gran parte de la aplicación se ha desarrollado basándose en la *programación orientada a objetos (POO)*, la cual implementa el concepto de clases, permitiendo que el código de programación desarrollado para la aplicación pueda ser utilizado por otras personas, con objeto de poder ofrecer una mayor potencialidad a la misma.
- El código fuente de *Académicos Web* es *independiente de la plataforma o sistema operativo* que se esté utilizando, esto quiere decir que el sistema puede funcionar perfectamente en servidores con sistemas operativos Linux o Windows.
- La aplicación se probó en una pequeña INTRANET, por tanto puede funcionar a nivel de una red de área local (LAN), o extenderla a Internet.

- Finalmente, el sistema cuenta dos tipos de manuales: el *Manual de Usuario* y el *Manual Técnico*. El manual del usuario contiene cada una de las pantallas con la descripción de su funcionamiento, lo que facilita al docente y usuarios el uso de la interfaz. El manual técnico contiene la documentación necesaria para que futuros programadores y diseñadores puedan ampliarlo.

RECOMENDACIONES

- *Académicos Web* se creó como un prototipo en su primera versión. Por tanto el diseño y desarrollo de la herramienta puede retomarse para incorporar nuevos cambios en base a la tecnología del momento. A su vez pudiese incorporarse el manejo de nuevos objetos interactivos como el FLASH, que ayuden a proporcionar un mejor realce a los cursos en línea.
- El Quiz de Ejercicios por el momento es solamente un cuestionario de preguntas con respuestas de selección múltiple. Este puede ser modificado de tal manera que incorpore objetos interactivos, de forma que, el alumno pueda desarrollar problemas ya sean de cálculos físicos o matemáticos en plena evaluación, dándole un mayor grado de dificultad a la misma.
- La unidad de tiempo del Quiz de Ejercicios es el segundo, esta puede modificarse de forma que involucre los minutos, con objeto de poder realizar evaluaciones más complejas y el ingreso en el tiempo para la evaluación sea más funcional.
- Por el momento el Quiz de Ejercicios es capaz de proporcionar el puntaje al final del test. Este puede mejorarse de forma que el sistema pueda administrar y almacenar las diferentes calificaciones por evaluación, con objeto de que el docente pudiese retomar una nota por alumno en base a los mismos.
- El editor posee las funciones necesarias y suficientes para poder crear un página web, este puede mejorarse tanto para darle una mejor presentación al mismo, como incorporársele nuevas funciones con objeto de construir una clase más atractiva.

- Puede diseñársele al Administrador de la aplicación una consola especial para la administración de los alumnos, docentes y materias vigentes en el sistema, de forma que le facilite dar privilegios sobre la aplicación a los docentes y alumnos, así como también poder asignar las materias que se encuentren activas, según el ciclo de estudio.
- La base de datos que se utilizó es MySQL, no obstante es posible que en el futuro se pueda cambiar con objeto de utilizar recursos que no tiene MySQL, como por ejemplo integridad referencial o manejo de procedimientos almacenados, utilizando para ello SyBase para Linux ó Sybase SQL AnyWhere para Windows 95, ya que también estas gozan de licencias libres.

VENTAJAS Y/ O BENEFICIOS

- El *Manual de Usuario de Académicos Web* permite que los usuarios utilicen de forma eficientemente la interfaz, ya que incluye pantallas y ejemplos para facilitar la interacción del docente con la herramienta, de igual modo posee un *Manual Técnico* que contiene todos los programas fuentes que se utilizaron para la creación del mismo. Esto con el objetivo de facilitar las posteriores modificaciones o ampliaciones del sistema.
- Dado que las aplicaciones en Internet son una opción más para el profesor y los estudiantes, la Universidad Don Bosco quiere formar parte de esta nueva era de la computación, por ello se considera que esta herramienta fomentará en los docentes el interés por aprender e incursionar en lo relacionado al mundo del Internet. Permitiendo también que el cúmulo de conocimientos de los docentes se almacene en una base de datos donde pueda ser consultado en cualquier momento por los estudiantes y enriquecido a través de las actualizaciones y/o modificaciones.
- El desarrollo de la codificación (programación) de la interfaz hace uso de características de Programación Orienta a Objetos (POO), implementado mediante ello el concepto de clases¹¹ que facilitan la modificación a la estructura de la aplicación sin necesidad de comprender todo el código fuente, basta con corregir este conjunto de objetos utilizados por la clase (los cuales también son utilizados en el programa principal de la interfaz) permitiendo de esta forma la reutilización del código, concepto que proporciona un ahorro en tiempo de desarrollo, aumentando la producción y reduciendo tiempos de programación.

¹¹ CLASE OBJETO que en la Programación Orientada Objetos es un conjunto de objetos con una estructura común.

BIBLIOGRAFIA

- **ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION
SEGUNDA EDICION**

McGraw- Hill

James A. Senn

Segunda Parte Análisis y Determinación de Requerimientos

Pág. 133, 177, 178

- **INGENIERIA DEL SOFTWARE UN ENFOQUE PRACTICO
CUARTA EDICION**

McGraw-Hill

Roger S. Pressman

Capítulo 12. Modelado del Análisis

Pag. 208,215,220,221,222

- **Direcciones de Internet**

[http:// bases.colnodo.apc.org](http://bases.colnodo.apc.org) Sitio de Colnodo dedicado a Interfaces de consulta para bases de datos en Internet. Tel. 232 4246 Bogotá Colombia.

<http://www.cybercursos.net/> Sitio que contiene diferentes Cursos por Internet.

<http://www.mysql.com> Sitio de Mysql, contiene toda la información referente a MySQL y permite grabar en disco el manual completo de MySQL.

<http://www.linux.com> Sitio que permite bajar una versión libre del sistema operativo de Linux.

<http://www.ictnet.es/esp> Foros de consulta, para hacer preguntas de determinados temas, permitiendo recolectar diferente información.

<http://www.torrealday.cjb.net/> Glosario de términos de Internet.

<http://www.discapnet.es/graficos/documentacion/glosario/index.asp>

Glosario técnico de Internet.

<http://www.nuestro.elsavador.com>

Sitio de Internet, que permite la generación de sitios Web, el cual es cortesía del Diario de Hoy.

<http://www.redestb.es/javaaula/cursjava.htm>

Sitio que permitió recolectar información sobre Java

<http://html.programacion.net/doc/js.htm>

Sitio que ayudo a conocer más sobre la programación de HTML, para interactuar con PHP3, para la generación del sitio de Académicos Web.

<http://www.ciberaula.com/cursos.html>

Este es una ciberaula, la cual ayudo a comparar y tener una idea de lo que es un curso en línea.

<http://www.pangea.org/pacoc/manuales/manual3.htm>

Diferentes manuales, para la programación de Html, Php3, y MySQL

<http://geneura.ugr.es/~maribel/php/#funcionesPHP>

Contiene diferentes funciones para comprender la programación de Php3

<http://www.find-script.com/scripts/PHP3/>

Muestra segmentos de códigos o scripts, para utilizarlos en la elaboración de programas.

GLOSARIO TECNICO

ALGORITMO Regla o proceso a seguir para realizar una tarea o llegar a la solución de un problema.

APPLET Pequeña aplicación accesible en un servidor Internet, que se transporta por la red, se instala automáticamente y se ejecuta *in situ* como parte de un documento Web.

ASCII American Standard Code for Information Interchange. Estándar Americano para Intercambio de Información. La tabla básica de caracteres ASCII esta compuesta por 128 caracteres incluyendo símbolos y caracteres de control. Existe una versión extendida de 256 caracteres.

BENCHMARK Programa especialmente diseñado para evaluar el rendimiento de un sistema.

BIT Binary Digit. Dígito Binario. Unidad mínima de información, puede tener dos estados "0" ó "1".

Bps (Bits per Second). Medida que representa la rapidez con que la información es transferida de una computadora a otra. Un módem de 28.8 Kbps. Es capaz de transferir 28.800 bits por segundo.

BYTE Unidad de medida de la cantidad de información en formato digital. Usualmente un byte consiste de 8 bits. Un bit es un cero (0) o un uno (1). Esa secuencia de números(byte) puede simbolizar una letra o un espacio (un carácter). Un kilobyte(KB) son 1024 bytes y un Megabytes(MB) son 1024 Kilobytes. (Ver también: Bit).

CD Compact Disc. Disco Compacto. Disco Optico de 12 cm de diámetro para almacenamiento binario. Su capacidad "formateado" es de 660 MB. Usado en principio para almacenar audio. Cuando se usa para almacenamiento de datos genéricos es llamado CD-ROM.

CGI Common Gateway Interface. Interface de Acceso Común. Programas usados para hacer llamadas a rutinas o controlar otros programas o bases de datos desde una página Web. También pueden generar directamente **HTML**.

CHAT Charla. Ver **IRC**.

IRC Internet Relay Chat. Canal de Chat de Internet. Sistema para transmisión de texto multiusuario a través de un servidor IRC. Usado normalmente para conversar on-line también sirve para transmitir ficheros.

IN SITU En el momento.

JAVA Lenguaje de programación orientado a objeto parecido al C++. Usado en *WWW* para la telecarga y telejecución de programas en el ordenador cliente. Desarrollado por Sun microsystems.

JAVASCRIPT Programa escrito en el lenguaje script de Java que es interpretado por la aplicación cliente, normalmente un navegador (Browser).

LAN Local Area Network. Red de Área Local. Red de ordenadores de reducidas dimensiones. Por ejemplo una red distribuida en una planta de un edificio.

LINK Enlace. Unión. Se llama así a las partes de una página WEB que nos llevan a otra parte de la misma o nos enlaza con otro servidor.

MODEM Modulator/Demodulator. Modulador/Demodulador. Dispositivo que adapta las señales digitales para su transmisión a través de una línea analógica. Normalmente telefónica.

Navegador Aplicado normalmente a programas usados para conectarse al servicio *WWW*.

PAQUETE Cantidad mínima de datos que se transmite en una red o entre dispositivos. Tiene una estructura y longitud distinta según el protocolo al que pertenezca. También llamado TRAMA.

PERL Lenguaje para manipular textos, ficheros y procesos. Con estructura de script. Desarrollado por Larry Wall, es multiplataforma ya que funciona en *UNIX*.

POP Post Office Protocol. Protocolo de Oficina de Correos. Protocolo usado por ordenadores personales para manejar el correo sobre todo en recepción.

ROOT Raíz. En sistemas de ficheros se refiere al directorio raíz. En Unix se refiere al usuario principal.

SQL Structured Query Language. Lenguaje de Petición Estructurada. Lenguaje para base de datos.

ANEXOS



**ENCUESTA PARA DETERMINAR EL ACCESO A
INTERNET POR PARTE ALUMNOS DE
INSTITUCIONES UNIVERSITARIAS**

La presente encuesta es realiza por alumnos de la Universidad Don Bosco para presentar Trabajo de Graduación titulado **"Elaboración de una interfaz automática para la creación de cursos académicos vía Web"**. El objetivo de la encuesta es determinar el grado o la facilidad de acceso a Internet por parte de los alumnos de Centros Educativos de nivel superior del área metropolitana de San Salvador.

Así mismo saber si conocen lo que son Cursos en Línea (Cursos Online), y que opinan de ellos y si ha recibido alguno en particular.

Por su valiosa colaboración Gracias.

1. ¿Qué carrera estudia? _____

2. ¿Tiene acceso a Internet? SI ☐ (Vaya a pregunta # 4) NO ☐ (Vaya a pregunta # 3)

3. ¿Que dificultades tiene para acceder a Internet?
 - ☐ La Universidad no proporciona este servicio
 - ☐ No tengo computadora
 - ☐ Acceso restringido a la Línea telefónica, por su alto costo
 - ☐ No tengo línea telefónica
 - ☐ Falta de Hardware (Módem) para efectuar la conexión a Internet Otros _____

4. ¿Desde que lugar accesa a Internet?
 - ☐ Lugar de Trabajo
 - ☐ Centro de Estudios
 - ☐ Casa
 - ☐ Ciber-café
 - Otros _____

5. ¿Qué facilidad le proporciona su Universidad para acceder a Internet?
 - ☐ Por laboratorio, por materias que incluyan el uso del centro de computo
 - ☐ Por horas establecidas sin costo
 - ☐ Por pagos adicionales de uso

6. ¿Sabe que son cursos en línea, o Cursos Online?
 - SI ☐ (Vaya a pregunta # 7)
 - NO ☐ (Vaya a pregunta # 11)

7. ¿Dónde ha escuchado sobre ellos?
 - ☐ Centro de estudios
 - ☐ Televisión
 - ☐ Trabajo
 - ☐ Revistas
 - ☐ Periódicos
 - Otros _____

8. ¿Que ventajas le encuentra? _____



**ENCUESTA PARA DETERMINAR EL ACCESO A
INTERNET POR PARTE ALUMNOS DE
INSTITUCIONES UNIVERSITARIAS**

9. ¿Ha visitado algún Curso en Línea (Curso Online)?

SI ☐ NO ☐

10. ¿Cuál es la dirección y de que trataba el sitio del Curso Online?

CURSOS ONLINE: Es un espacio abierto para la reflexión educativa que cuenta con una variada información relacionada con el hecho educativo en sus distintas manifestaciones. Es un medio eficaz para distribuir información rápidamente, la cual tiene la influencia de la tecnología y el conocimiento de los expertos de los cursos que se imparten, a través de Internet.

11. ¿ Le gustaría tener un Curso en Línea en su centro educativo?

SI ☐ NO ☐

Por qué? _____



ENCUESTA SOBRE EL MANEJO DE INTERNET POR PARTE DE DOCENTES DE INSTITUCIONES EDUCATIVAS

Esta encuesta es desarrollada por alumnos de la carrera de Ingeniería en ciencias de la computación, para presentar Trabajo de Graduación, sobre el tema "Elaboración de una interfaz automática par al creación de cursos académicos vía Web". Dicha encuesta tiene como objetivo, determinar el conocimiento de los docentes sobre el uso de Internet como herramienta de apoyo para la enseñanza, así como el grado de conocimiento que los docentes tienen en cuanto al diseño de aplicaciones para el Web.

Agradecemos la colaboración que nos brindan con la resolución de esta encuesta, aportándonos valiosa información para la elaboración de este trabajo de graduación.

DATOS DE CLASIFICACION

1. NIVEL DE EDUCACIÓN AL QUE IMPARTE CLASES

☐ Técnico superior

☐ Humanidades

☐ Ingeniería

☐ Maestría

☐ Licenciatura

☐ Otros _____

2. AREA DE CONOCIMIENTO EN QUE DESARROLLA SU LABOR DOCENTE

☐ Matemática-Física

☐ Área especializada de Ingeniería

☐ Ciencias – Humanidades

☐ Área básica de Ingeniería

☐ Educación

☐ Otros _____

PREGUNTAS

1. ¿Tiene acceso a Internet desde su casa?

☐ Sí (Vaya a pregunta 2)

☐ No (Vaya a pregunta 4)

Porqué? _____

2. ¿Qué uso le dá actualmente a Internet?

☐ Búsqueda de información de general y entretenimiento

☐ Comunicación vía correo electrónico

☐ Platicar en el Chat

☐ Transacciones bancarias o económicas

☐ Comercio electrónico (comprar libros, CD de Software, Música, etc.)

☐ Actualizaciones de Software (Bajar programas)

☐ Consultas a Bibliotecas

☐ Lecturas de revistas y periódicos

☐ Resolver test o exámenes de cursos específicos.

☐ Foros de comunicación



**ENCUESTA SOBRE EL MANEJO DE INTERNET POR PARTE DE
DOCENTES DE INSTITUCIONES EDUCATIVAS**

3. ¿De los usos antes mencionados, cuáles considera que apoyan el proceso de aprendizaje-enseñanza?, ¿Por qué?

4. ¿ Considera que los cursos en Internet pueden mejorar el aprendizaje de sus alumnos?

- ☐ Sí ¿Por qué? _____
- ☐ No ¿Por qué? _____

5. ¿ Puede Ud. Diseñar curso en Internet?

- ☐ Sí (pase a la pregunta 7) ☐ No (pase a la pregunta 6)

6. ¿Cuál es la dificultad?

- ☐ No sabe que son cursos en Internet
- ☐ No conoce nada sobre el diseño de páginas para Internet
- ☐ Nunca he trabajado con un lenguaje de programación
- ☐ No existe una herramienta que me facilite este trabajo
- ☐ No tengo acceso al uso de una computadora

Otros

7. Marque con una "X", el uso que le daría a una herramienta o asistente que le permita diseñar Ud. Mismo cursos o clases para ser colocadas en Internet

- ☐ Pondría ejercicios para evaluar a los alumnos
- ☐ Sesiones de discusión, sobre puntos de clases, utilizando el chat.
- ☐ Digitalización de la información de una clase para que esté a disposición del alumno.
- ☐ Envío de información a los alumnos vía correo electrónico.
- ☐ Promover la investigación en Internet, para temas específicos de clases.

Otros



**ENCUESTA SOBRE EL MANEJO DE INTERNET POR PARTE DE
DOCENTES DE INSTITUCIONES EDUCATIVAS**

8. Marque cuales de estas herramientas conoce, para el diseño de páginas Web

☐ HTML

☐ FrontPage

☐ Word

☐ Plubisher

Otros

9. ¿Cuánto tiempo le dedica Ud. A la preparación de su clase?

☐ 2 – 4 Horas

☐ 6 – 8 Horas

☐ 4 – 6 Horas

☐ Más de 8 Horas

10. ¿Ha vistado Ud algún curso en Línea en Internet?

☐ Sí (pase a la pregunta 11)

☐ No (pase a la pregunta 12)

11. ¿Cual es la dirección, sobre que trataba el Curso y que opina de el?

12. Comentarios o sugerencias.

ANEXO 3

GUIA DE INSTALACION DE UN SERVIDOR WEB

Nota:

*En nuestro sistema el nombre del host, para poder verificar la base de datos, se llama **princesa**, en caso de no querer utilizar este nombre de host, cambiarlo en todos los programas el nombre del mismo al que guste y configurarlo de esa forma en la base de datos mysql.*

Pasos a seguir para instalar un servidor apache en windows con módulos php.

1. Instalar el software phptriadsetup para windows. (Esto instala apache, php y mysql)

Detalle en la instalación de apache y php (La mayoría debe aparecer configurado, solo será necesario verificar. El único paso a modificar será el que nos indica la línea: #ServerName new.host.name en el archivo http.conf)

APACHE

Editamos el fichero de configuración **http.conf** que se halla en C:\Apache\conf\

Buscamos la línea donde aparece:

#ServerName new.host.name

Quitamos el comentario (#) y lo reemplazamos por lo siguiente:

ServerName http://princesa

**** En el caso de querer conservar ese nombre de host, de lo contrario colocar el que guste ****

Indicamos el directorio de PHP:

ScriptAlias /php "C:\apache\php"

Definimos la extensión de los script PHP:

AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3 .php
AddType application/x-httpd-php3 .phtml

Y asignamos la aplicación para las extensiones PHP:

Action application/x-httpd-php3 "/php/php.exe"

Por defecto los ficheros que son accesibles desde el navegador se encuentran en la carpeta **htdocs** del directorio de Apache, pero la podemos cambiar:

```
DocumentRoot "C:\www"  
<Directory "C:\www">  
.....  
</Directory>
```

PHP

Para configurar PHP, primero buscamos el archivo **php3.ini-dist** el cual se encuentra en C:\apache\php y lo renombramos a **php.ini**, después lo editamos y le hacemos los siguientes cambios:

Buscamos la expresión "**extension_dir**" y la cambiamos por:
extension_dir = C:\apache\php

Para añadir el soporte para MySQL buscar la línea:
; extension = php3_mysql.dll

Reemplazarla por:

extension = php3_mysql.dll

Copiar el archivo **php3.ini** en "**C:\windows**"

Ejecución de los programas:

Pues bien, ya solo nos queda arrancar los programas:

C:\Apache\apache.exe
C:\mysql\bin\mysqld.exe

#Para la versión shareware
C:\mysql\bin\mysqld-shareware.exe

También podemos arrancar el servidor Apache desde el menú de inicio:
Inicio->Programas->phptriad for windows->Start Apache

Para comprobar nuestra instalación crear un fichero llamado test.php3 con la siguiente línea:

```
<?php phpinfo() ?>
```

Colócalo en el directorio de documentos de Apache y llámalo desde el navegador. Si lo hemos hecho todo bien nos saldrá una página con todas las variables de PHP.

NOTA:

Cabe destacar que lo que hemos hecho es una **instalación básica**, por lo que recomendamos leer los manuales de las distintas aplicaciones para obtener más detalles sobre la instalación de éstas.

2. Ingresar a phpMyAdmin y configurar en la base de datos mysql, las tablas user, host y db.

La estructura de MySQL

Para cada base de datos que nosotros creamos, MySQL crea un directorio con el nombre que le hemos asignado a la base de datos. Dentro de este directorio, por cada tabla que definamos MySQL va a crear tres archivos: **mitabla.ISD**, **mitabla.ISM**, **mitabla.frm**

El archivo con extensión **ISD**, es el que contiene los datos de nuestra tabla, el **ISM** contiene información acerca de las claves y otros datos que MySQL utiliza para buscar datos en el fichero **ISD**. Y el archivo **frm** contiene la estructura de la propia tabla.

Dado que las bases de datos de MySQL son simples ficheros de un directorio, para realizar copias de seguridad, podremos utilizar las herramientas de compresión que habitualmente usamos en nuestro sistema y luego copiarlo a otro lugar, o simplemente esto último.

Seguridad

Como comentamos anteriormente, todo el sistema de permisos MySQL lo guarda en una base de datos llamada mysql, la cual se compone de cinco tablas: **host**, **user**, **db**, **tables_priv**, **columns_priv**.

La tabla **user** contiene información sobre los usuarios, desde que máquinas pueden acceder a nuestro servidor MySQL, su clave y de sus diferentes permisos. La tabla **host** nos informa sobre que máquinas podrán acceder a nuestro sistema, así como a las bases de datos que tendrán acceso y sus diferentes permisos. Finalmente, las tablas **db**, **tables_priv**, **columns_priv** nos proveen de un control individual de las bases de datos, tablas y columnas (campos).

Tabla user

CAMPO	TIPO	PERMISOS
Host	char(60)	
User	char(16)	
Password	char(16)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N

Drop_priv	enum('N','Y')	N
Reload_priv	enum('N','Y')	N
Shutdown_priv	enum('N','Y')	N
Process_priv	enum('N','Y')	N
File_priv	enum('N','Y')	N
Grant_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla host

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
Db	char(32)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
Grant_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla db

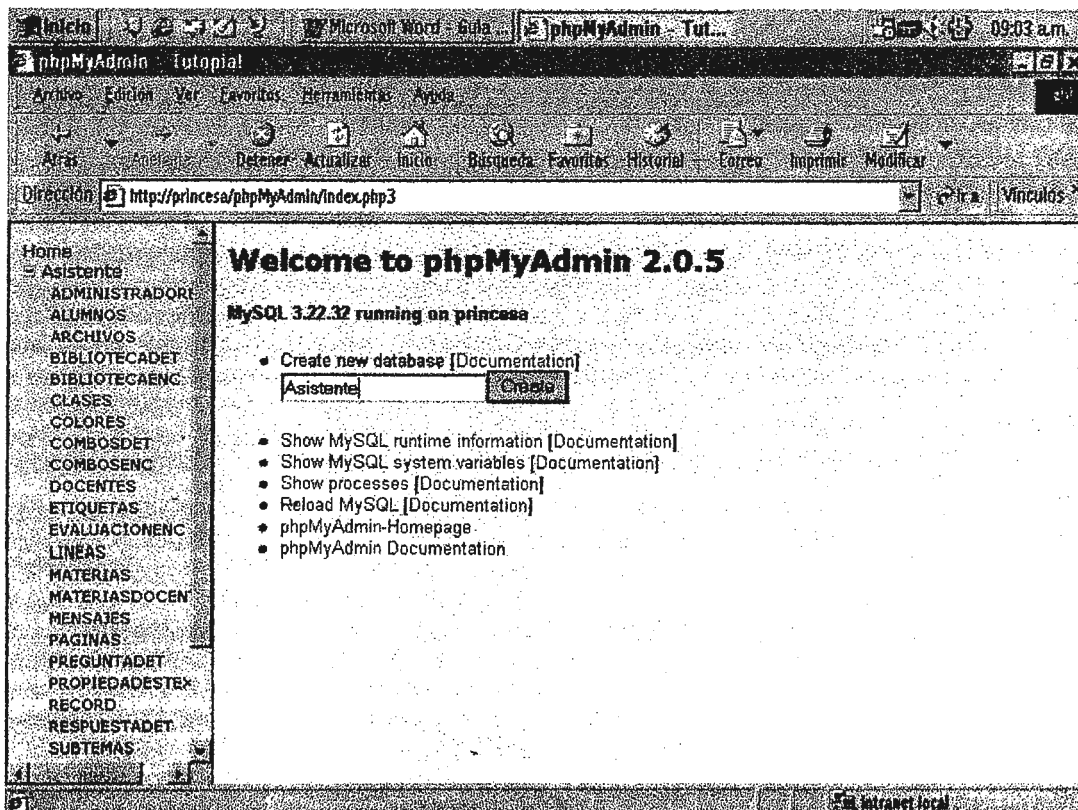
CAMPO	TIPO	POR DEFECTO
Host	char(60)	
Db	char(32)	
User	char(16)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

He aquí una breve descripción de los diferentes permisos:

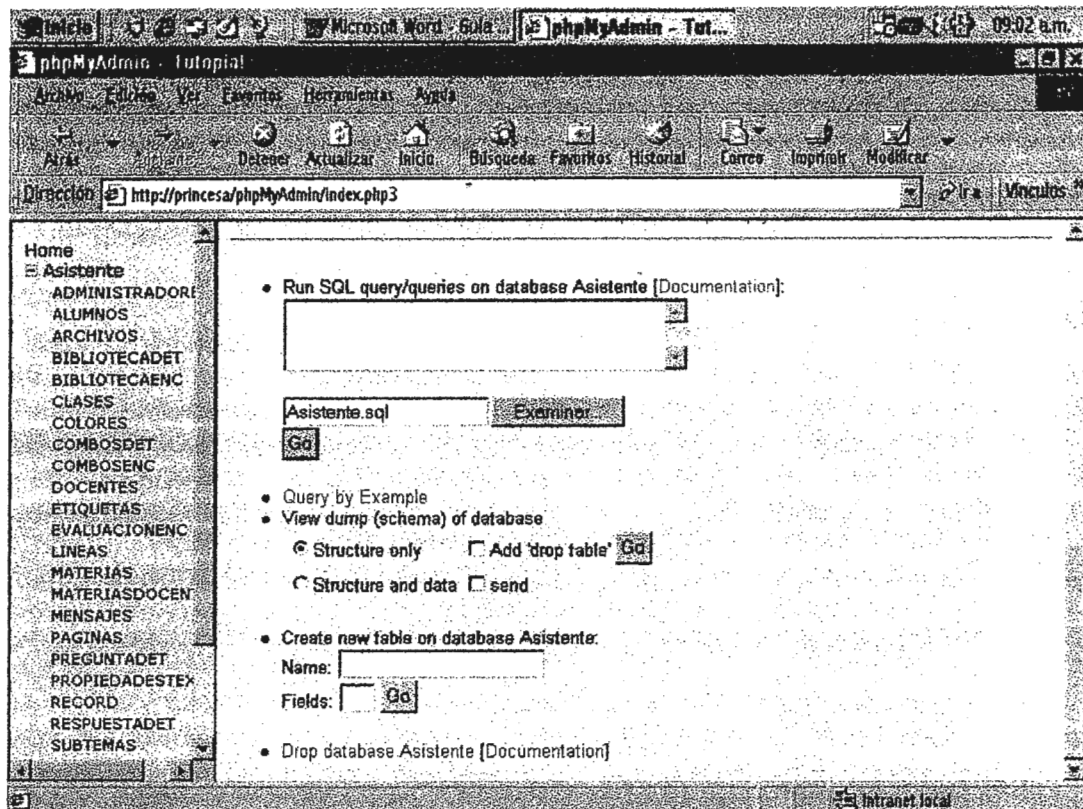
- **Select_priv:** Permite utilizar la sentencia SELECT
- **Insert_priv:** Permite utilizar la sentencia INSERT
- **Update_priv:** Permite utilizar la sentencia UPDATE
- **Delete_priv:** Permite utilizar la sentencia DELETE
- **Create_priv:** Permite utilizar la sentencia CREATE o crear bases de datos
- **Drop_priv:** Permite utilizar la sentencia DROP o eliminar bases de datos
- **Reload_priv:** Permite recargar el sistema mediante *mysqladmin reload*
- **Shutdown_priv:** Permite parar el servidor mediante *mysqladmin shutdown*
- **Process_priv:** Permite manejar procesos del servidor
- **File_priv:** Permite leer y escribir ficheros usando comando como SELECT INTO OUTFILE y LOAD DATA INFILE
- **Grant_priv:** Permite otorgar permisos a otros usuarios
- **Index_priv:** Permite crear o borrar índices
- **Alter_priv:** Permite utilizar la sentencia ALTER TABLE

Si dejamos en blanco los campos **user**, **host** o **db**, haremos referencia a cualquier usuario, servidor o base de datos. Conseguiremos el mismo efecto poniendo el símbolo % en el campo.

3. Crear la base de datos con la herramienta phpMyAdmin. (Esta es la herramienta de administración de bases de datos mysql escrita en php)



4. Correr el script Asistente.sql para restablecer la base de datos de nuestra aplicación en el nuevo host, desde la consola phpMyAdmin.



5. Copiar en httdocs o el directorio de publicación que asigmo las carpetas:

- **Html**
- **Php3**
- **PsImagen**
- **PsVideo**
- **PsSonido**
- **PsInteractividad**
- **PsVrml**

Estas contienen los programas a utilizar para nuestra aplicación.

6. En el Explorador digitar:

<http://princesa/Html/>

Para ingresar a la página principal de la aplicación. (Recuerde que el nombre del host puede cambiar).

ANEXO 5

PRESUPUESTO COSTO/BENEFICIO DEL PROYECTO

HARDWARE		
Cantidad	Equipo	Costo en Colones
1	SERVIDOR <ul style="list-style-type: none"> - PENTIUM INTEL III DE 533 Mhz - 72 Mb DE Memoria RAM - 10 Gb de Disco Duro - Multimedia con CD-ROM 4 X - MODEM 	6,000.00
1	CLIENTES <ul style="list-style-type: none"> - PENTIUM III DE 100 Mhz - 32 Mb de Memoria RAM - 1 Gb de Disco Duro - Multimedia con CD-ROM 2 X Max - MODEM 	5,500.00
1	<ul style="list-style-type: none"> - IBM 100 Mhz - 72 Mb de Memoria RAM - 1 Gb de Disco Duro - Multimedia con CD-ROM de 4 X Max - MODEM 	7,750.00
1	<ul style="list-style-type: none"> - PENTIUM INTEL DE 550 Mhz. - 128 Mb de Memoria RAM - 10.2 Gb de Disco Duro - Multimedia con CD-ROM de 52 X Max - MODEM 	6,350.00
3	REGULADORES DE VOLTAJE 350.00 (C/U)	1,050.00
1	IMPRESORA CANON BJC-3000	3,500.00
1	IMPRESORA EPSON STYLUS COLOR 600	2,550.00
4	TINTAS (NEGRO Y COLOR) CANON BJC-3000	1,750.00
2	TINTA (NEGRO Y COLOR) EPSON STYLUS 600	750.00
3	TARJETAS DE RED ENL 832-TX 10/100 M FAST ETHERNET NETWORK ADAPTER	510.00
2	CABLEADO (CALBE CROSSOVER)	160.00

	MOBILIARIO Y EQUIPO	
3	MESA PARA EQUIPO	2,200.00
4	SILLAS	475.00
	PAPELERIA (PAPEL EMPASTADO)	2,510.00
	SOFTWARE	
2	LICENCIAS DE WINDOWS 98	4,000.00
2	LICENCIAS DE OFFICE 97	6,000.00
	SERVICIOS	
	ENERGIA ELECTRICA (SEMANAL)	16.26
	AGUA (SEMANAL)	5.00
	TELEFONO (SEMANAL)	93.75
	GASTOS VARIOS (DIARIOS, REFRIGERIOS)	5,040.00
	RECURSO HUMANO	
	HORA POR PERSONA : 5 DIAS/SEMANA DE 8:00 P.M - 1:00 A.M COSTO ¢60.00 : ¢ 300.00 (x 5 DIAS)=¢1,500.00 FIN DE SEMANA DE 3:00 - 11:00 P.M COSTO ¢60.00: ¢480.00 (X 2 DIAS)= ¢960.00 COSTO SEMANAL: ¢2,460.00 (X 4 SEMANAS) COSTO MENSUAL: ¢9,840.00 (X 6 MESES)	59,040.00
	TOTAL	¢ 115,250.00

ANEXO 6

DIAGRAMA DE FLUJO DE DATOS DE LA INTERFAZ AUTOMATICA PARA LA CREACIÓN DE CURSOS ACADÉMICOS VIA WEB

NIVEL 0

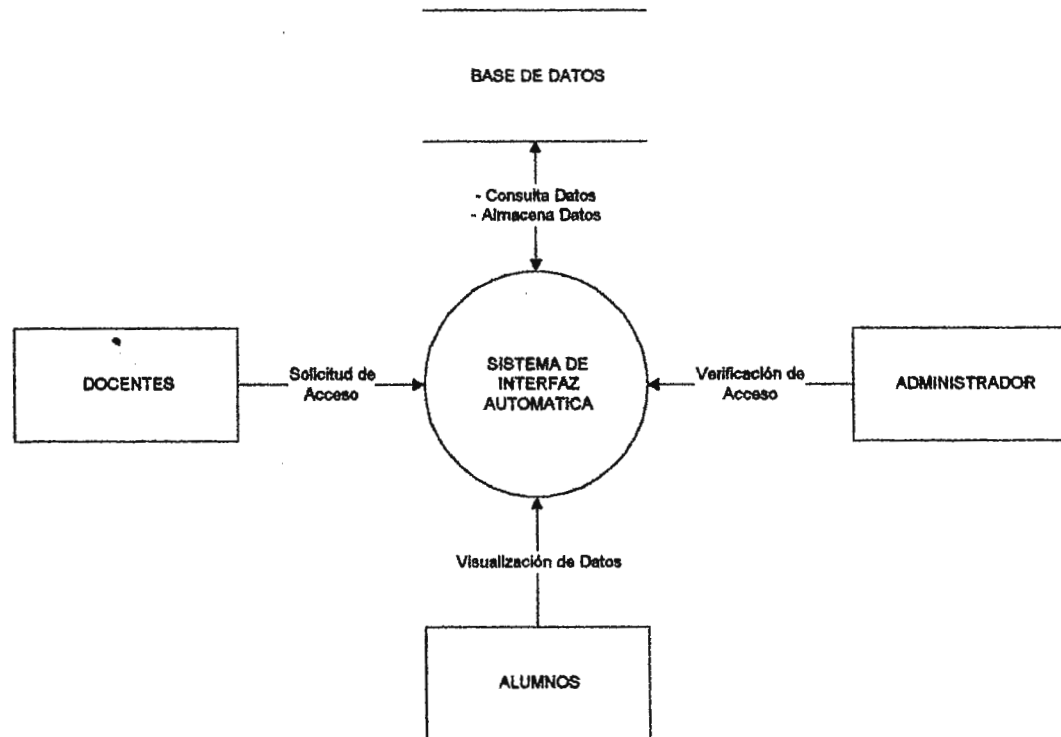


Diagrama de flujo de datos para el Sistema de Interfaz Automática, en su nivel más general, donde se detallan todas las entidades que forman parte de la Interfaz, y los procesos que deben realizar.

NIVEL 1

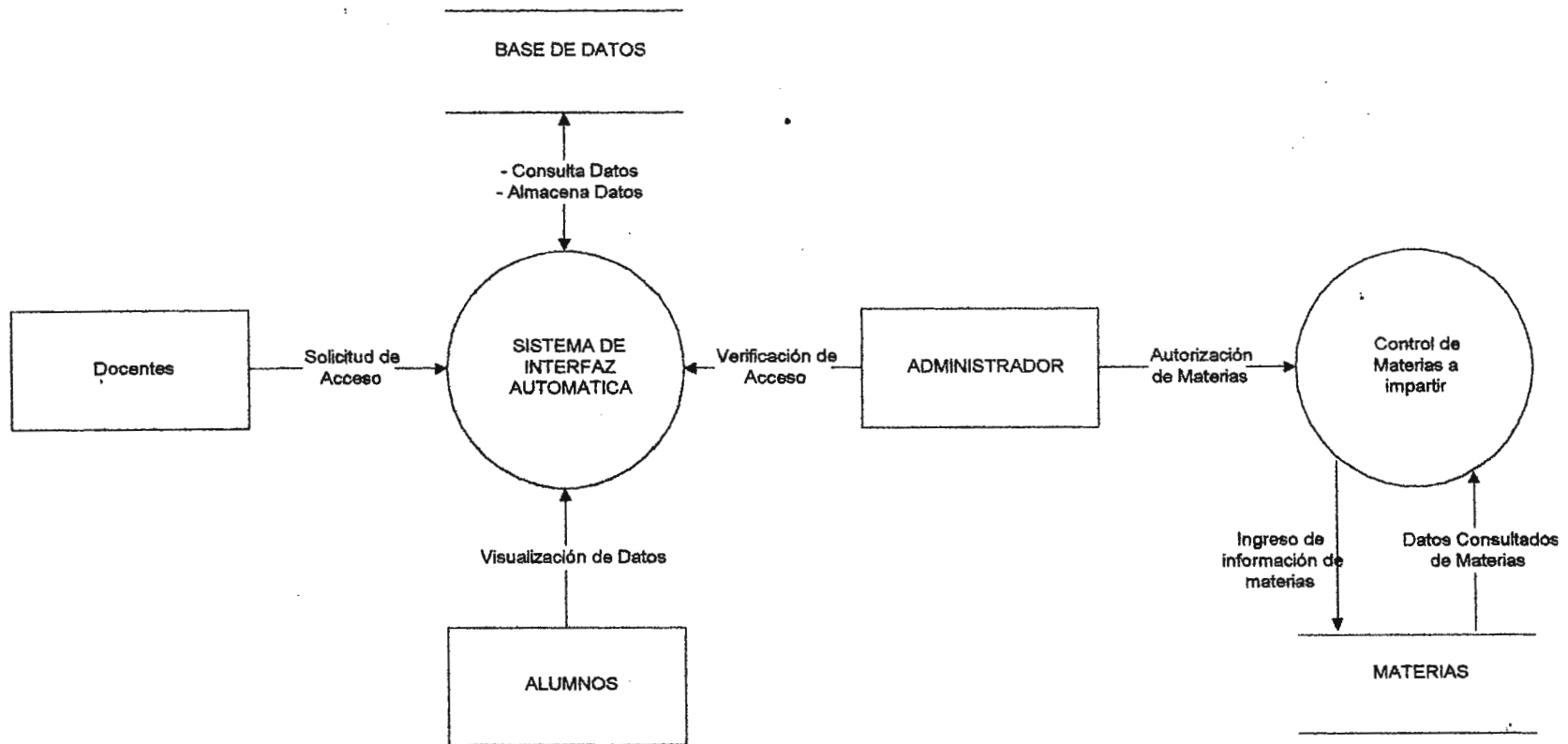
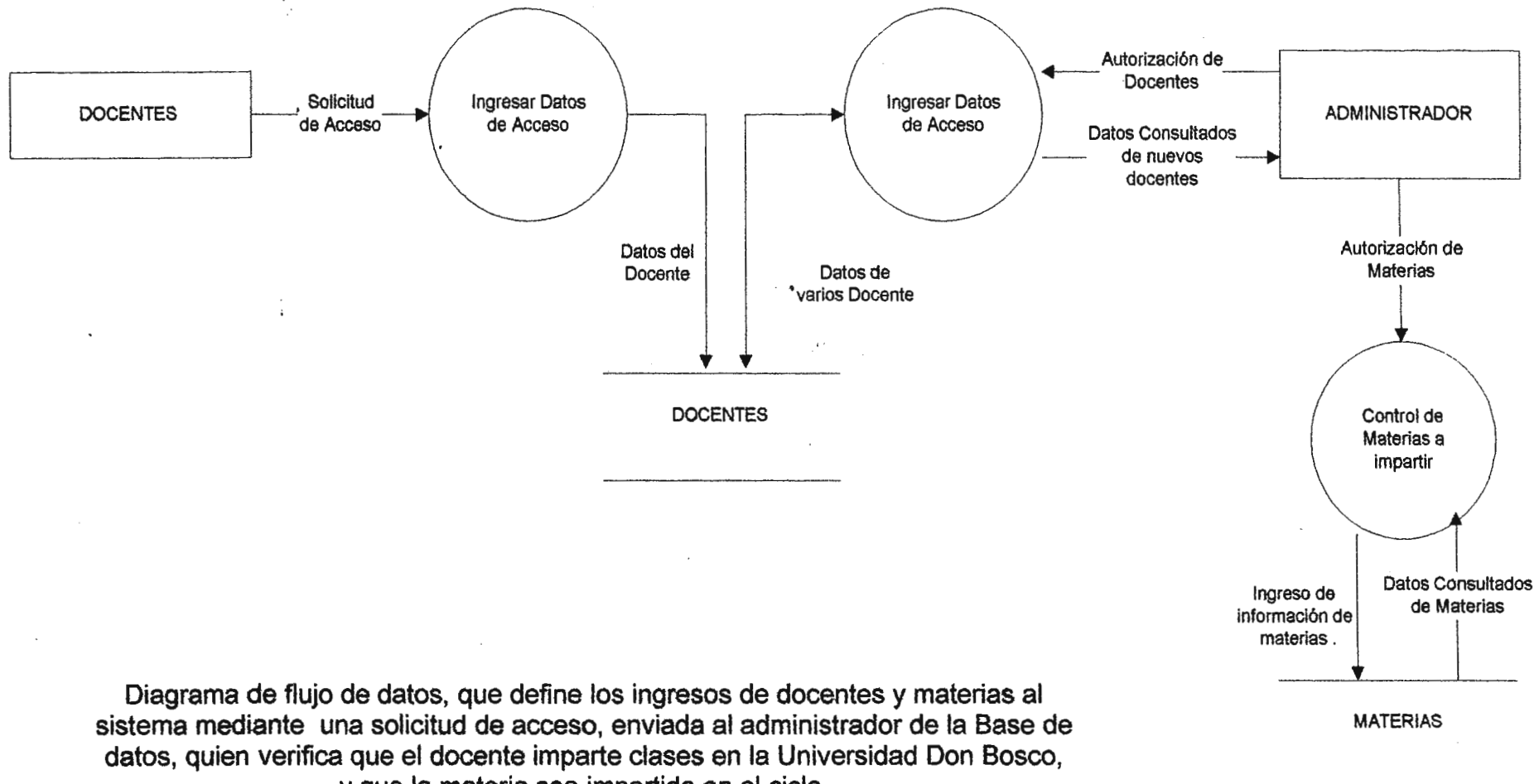


Diagrama de Flujo de datos, para los proceso de Control de materias a impartir, a cargo del Administrador de la Base de datos, el cual verifica el acceso al sistema de interfaz.

NIVEL 2



NIVEL 3

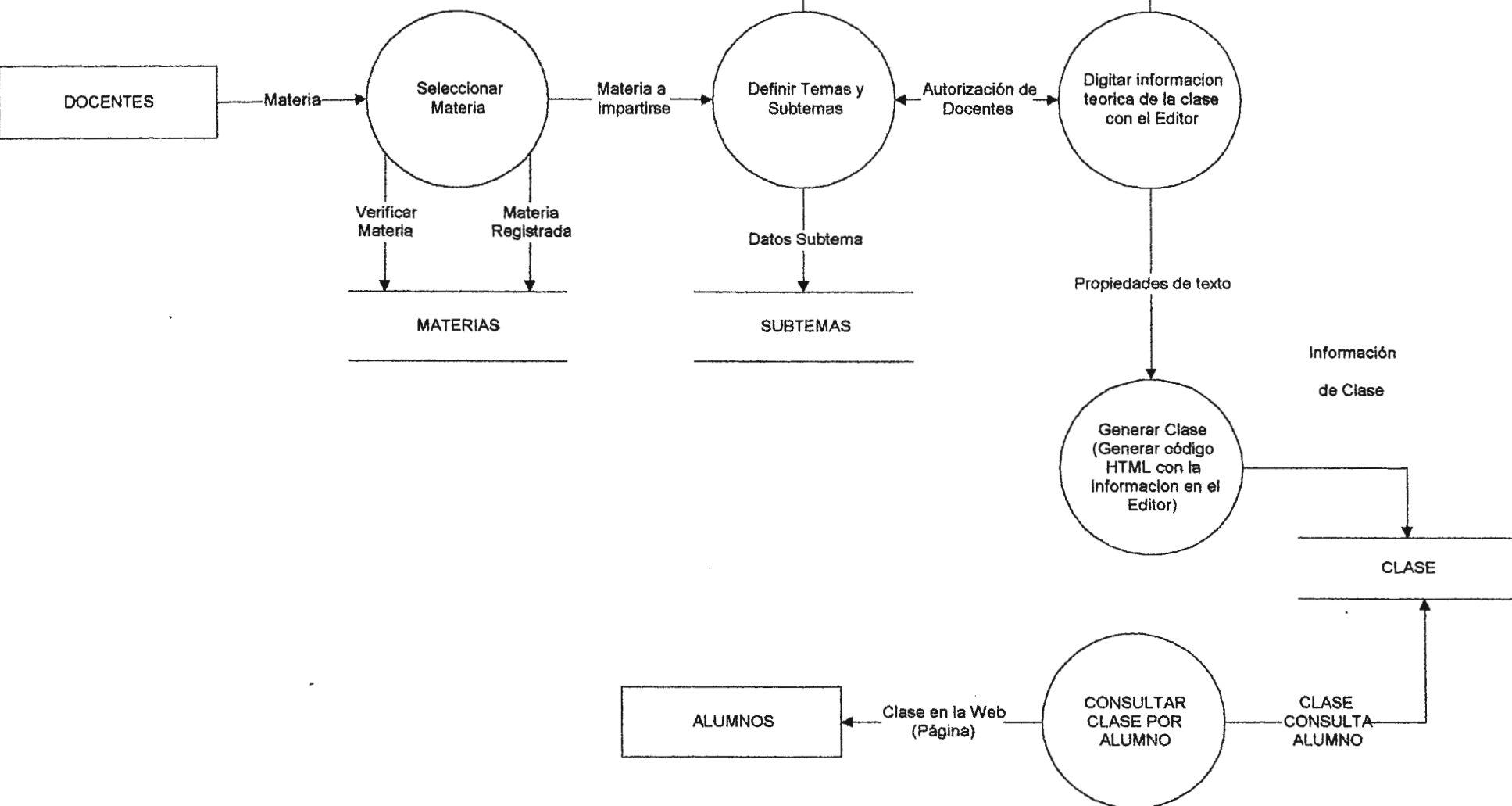
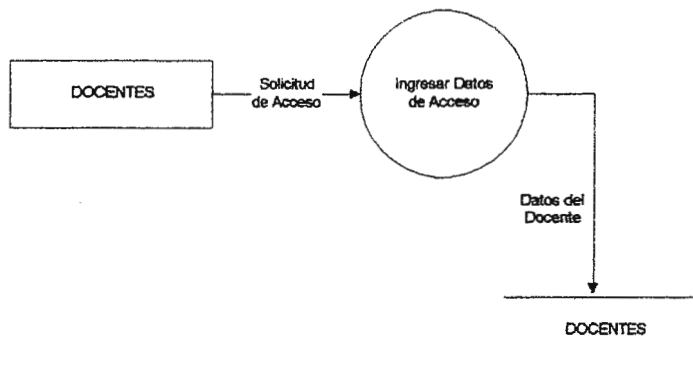


Diagrama de flujo de datos, que describe el proceso de creación de la clase, que va desde la selección de la materia, definición de temas y subtemas, digitalización del contenido de la clase en Editor de la aplicación, para luego generar la clase y almacenarla en la base de datos, para que pueda ser vista por sus alumnos.

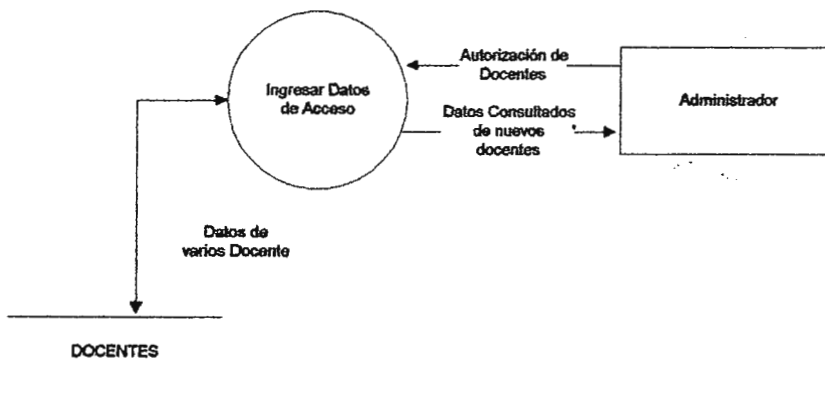
ANEXO 7

ESPECIFICACION DE PROCESOS DEL SISTEMA DE INTERFAZ AUTOMATICA PARA LA CREACION DE CURSOS ACADEMICOS VIA WEB



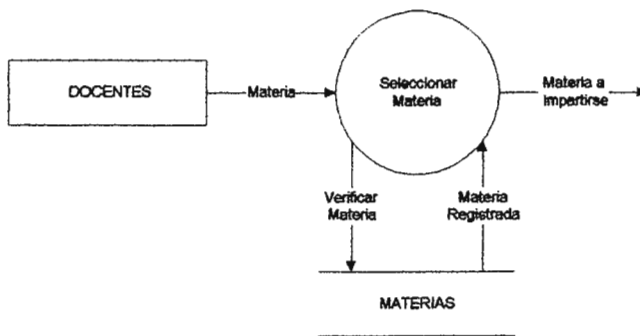
E.P. INGRESAR DATOS

En este proceso el docente debe de llenar un formulario con sus datos personales, para que sean almacenados en la base de datos en la tabla correspondiente a los Docentes.



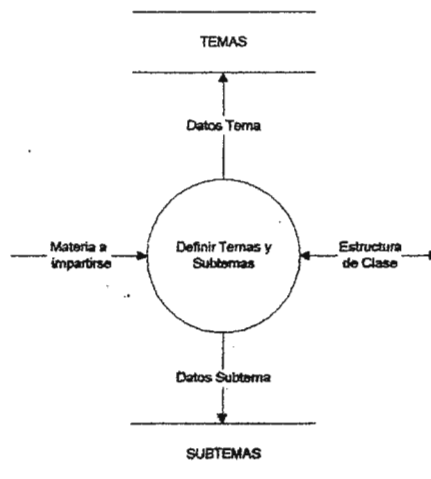
E.P. INGRESAR DATOS DE ACCESO

El docente ingresa los datos de solicitud de acceso, los cuales son consultados por el Administrador y luego de verificar que el docente esta registrado en la Base de Datos, envía una autorización al docente para acceder al Sistema de Interfaz.



E.P. CONTROL DE MATERIAS A IMPARTIR

El Administrador se encarga de autorizar a los docentes para desarrollar clases de las materias registradas a impartirse en cada ciclo, si la materia se encuentra en la tabla de materias de la Base de Datos, el administrador permite que los docentes desarrollen clases en el Web, de la materia seleccionada y solicitada.



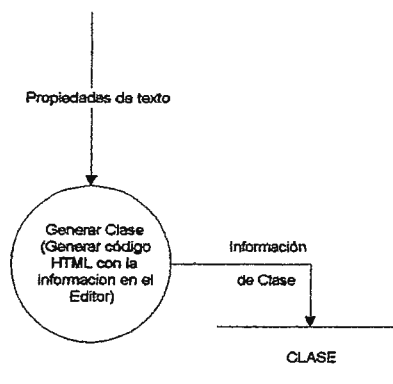
E.P. DEFINIR TEMAS Y SUBTEMAS

Al efectuar este proceso, se definen los temas y subtemas de la materia seleccionada por el docente, los cuales son almacenados en sus respectivas tablas, de la Base de Datos, con lo que se comienza con la estructura de la clase, la cual está lista para el siguiente proceso.



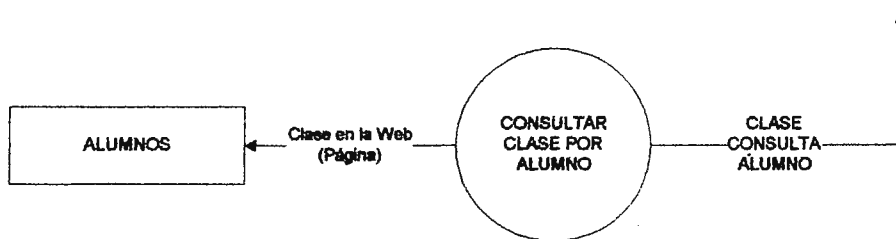
E.P. DIGITAR INFORMACIÓN TEORICA CON EL EDITOR

Luego de comenzar a definir los temas y subtemas, se procede con la estructuración de la clase, digitando en el editor, el contenido de la clase teórica a desarrollar, luego de esto se almacena la información y se envía esta al siguiente proceso



E.P. GENERAR CLASE (GENERAR CODIGO HTML)

Este proceso consiste en que cuando el docente ha terminado de digitar toda su información teórica, al hacer clic en el botón Generar clase, colocará de forma automática los comandos utilizados en la elaboración de páginas Web, las cuales luego de generadas serán almacenada en la tabla "Clase", para que posteriormente el alumno pueda consultarlas.



E.P. CONSULTAR CLASE POR ALUMNO

Este proceso consiste en presentar a los alumnos las diferentes clases creadas por sus maestros, quienes les piden que visiten el sitio y que resuelvan un cuestionario en la clase.

Manual de Usuario

ELABORACION DE UNA INTERFAZ AUTOMATICA PARA LA CREACION DE CURSOS ACADEMICOS VIA WEB

El manual de usuario es una guía que explica de manera clara y precisa cada una de las pantallas y su funcionamiento. Lo que proporcionara al usuario la facilidad para el uso de la interfaz.

Pantalla: Inicio de Sesión

Al ingresar a la aplicación el usuario visualiza la pantalla de inicio de sesión la cual esta dividida en dos partes

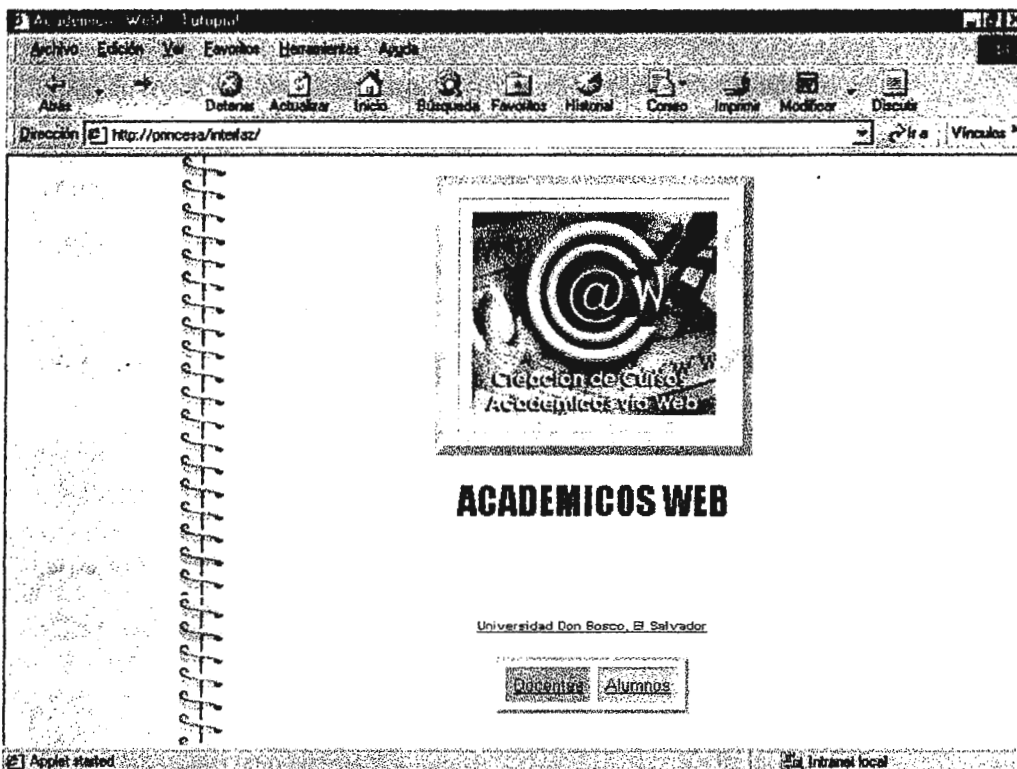
En la parte izquierda se presenta el menú principal el cual contiene las opciones de:

Menú Docentes: contiene las siguientes opciones :

- **Registrarse:** Esta opción permite al docente solicitar el acceso a la interfaz
- **Crear clase:** Esta opción permite al docente que elabore su clase definiendo cada uno de los elementos para desarrollarla
 - temas
 - subtemas
- desarrollo de la clase
- **Evaluación:** Esta opción permite que el docente desarrolle test sobre los temas desarrollados, para que después sean contestados por los alumnos.
- **Comentarios:** Esta opción permite al docente visualizar cada uno de los comentarios recibidos, los cuales son enviados por los alumnos.

Menú Alumnos: contiene las siguientes opciones:

- **Consultar las clases:** Las clases que son creadas por los docentes pueden ser visualizadas por los alumnos desde esta opción.
- **Realizar Evaluaciones :** Esta opción contiene cada uno de los test que el docente desarrolla en los cuales el alumno pone a prueba sus conocimientos.
- **Escribir comentarios a los docentes:** Esta opción es utilizada para enviar comentarios a los docentes a través de su correo electrónico.



Campos

Usuario: El usuario digita el nombre de usuario que tiene asignado

Password: Cada usuario posee una contraseña, y se digita en este espacio

La pantalla también presenta los botones de:

Entrar: Al darle clic en este botón se verifica si el usuario y contraseña son correctos, si es así es se presenta la pantalla de bienvenida.

Borrar: Limpia los campos de usuario y contraseña

LogOut: Al darle clic en este botón se cierra la sesión que esta activa

Pantalla: Inicio de Sesión, opción Registrarse

Si el usuario utilizara la aplicación por primera vez, debe de darle clic a la opción "Registrarse" que esta en el menú principal o en el área de inicio de sesión, Lo que le permitirá solicitar sus derechos como usuario.

The screenshot shows a Netscape browser window with the title "Interfaz de Entrada Academicos Web - Titulo". The address bar displays "http://pincasa/interfaz/interfaz.php3". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The toolbar contains icons for "Detener", "Actualizar", "Inicio", "Búsqueda", "Favoritos", "Historial", "Correo", "Imprimir", "Modificar", and "Discutir".

The main content area features a "REGISTRARSE" button at the top. Below it is a large rounded rectangle with the text "ACADEMICOS WEB" and "¡BIENVENIDO!". Underneath this are two input fields labeled "Entrada:" and "Contraseña:". At the bottom of the rectangle are a small "X" icon on the left and a checkmark icon on the right. Below the rectangle, the text "Universidad Don Bosco 2001" is visible.

The status bar at the bottom shows "Línea" on the left and "Internet local" on the right.

Pantalla: Registrarse

Cuando el usuario da clic en la opción registrarse, visualiza un formulario como el siguiente el cual debe de llenar un formulario con sus datos personales, los cuales son verificados por la persona encargada de darle la autorización. El formulario tiene la siguiente forma:

The screenshot shows a web browser window with the address bar displaying `http://princesa/Html/registrodocente.html`. The page title is "Programa de Docentes - Totalpad". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The toolbar contains icons for "Atrás", "Detener", "Actualizar", "Inicio", "Búsqueda", "Favoritos", "Historial", "Conexión", "Imprimir", "Modificar", and "Detener". The address bar also shows "Vínculos".

The main content area is titled "DATOS DEL DOCENTE" and contains a registration form with the following fields:

NOMBRE	<input type="text"/>
APELLIDOS	<input type="text"/>
E-MAIL	<input type="text"/>
USUARIO	<input type="text"/>
CONTRASEÑA	<input type="text"/>
CONFIRMAR CONTRASEÑA	<input type="text"/>
PROFESION	<input type="text" value="Ing. Computación"/>
CARGO	<input type="text"/>
DIRECCION	<input type="text"/>

The status bar at the bottom shows "Listo" and "Internet local".

Descripción de Campos:

Nombre y Apellido: El usuario digita su nombre y apellido

Email: Digita su dirección de correo electrónico

Usuario: Se digita un nombre de usuario el cual permitirá el inicio de la sesión

Contraseña: Se digita una contraseña la que permite el inicio de la sesión

Confirmar contraseña: Permite verificar si la contraseña es correcta

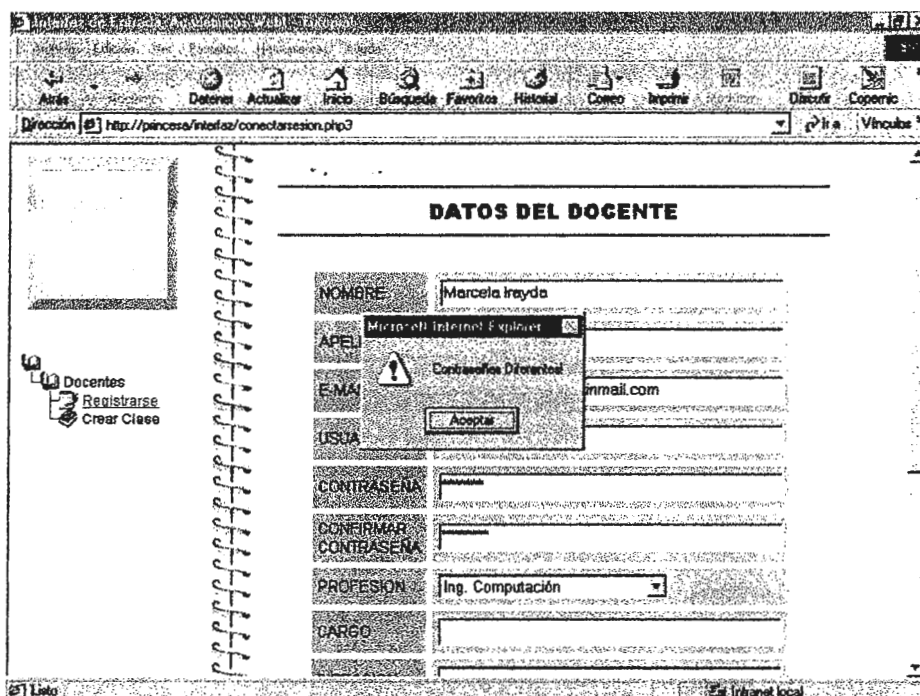
Profesión: Se especifica la actividad que la persona desarrolla

Cargo: Se define el cargo que desempeña el usuario

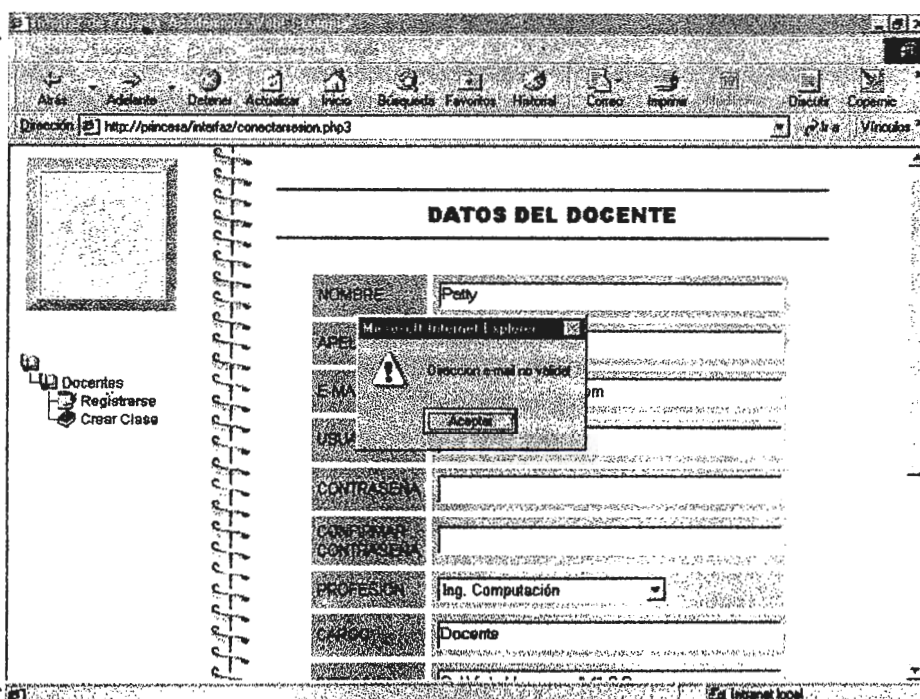
Dirección: El usuario digita la dirección donde reside

Teléfono y fax: Se digita el número de teléfono y fax

Cuando el usuario llena el formulario se le solicita una contraseña la cual debe de ser reconfirmada, si no coinciden se presenta una ventana con el mensaje siguiente:

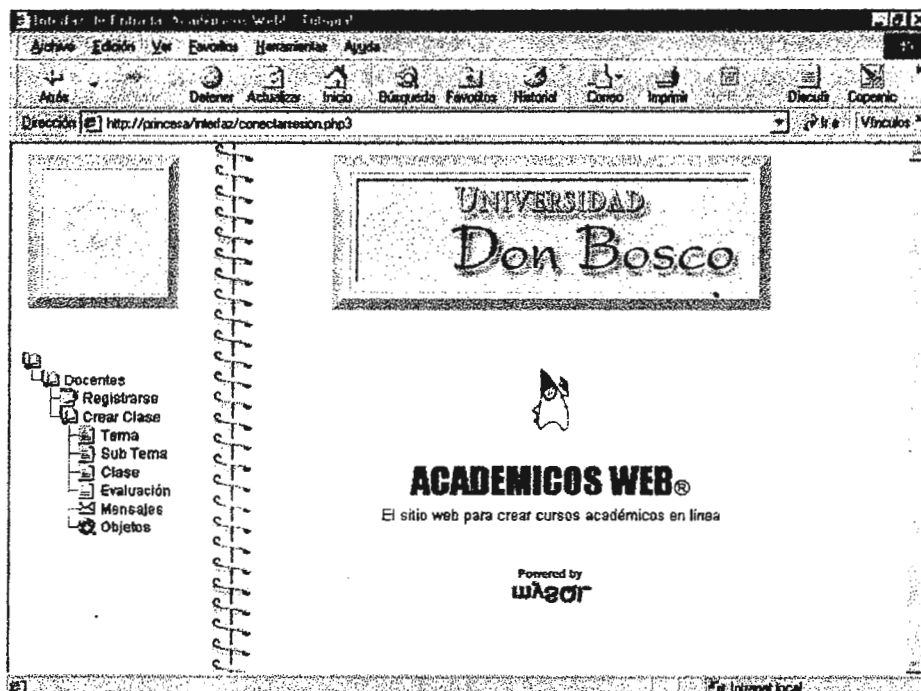


También se puede visualizar la validación de la dirección de correo, la cual no es valida si el usuario no digita @ dentro de ella, el mensaje de advertencia es el siguiente:

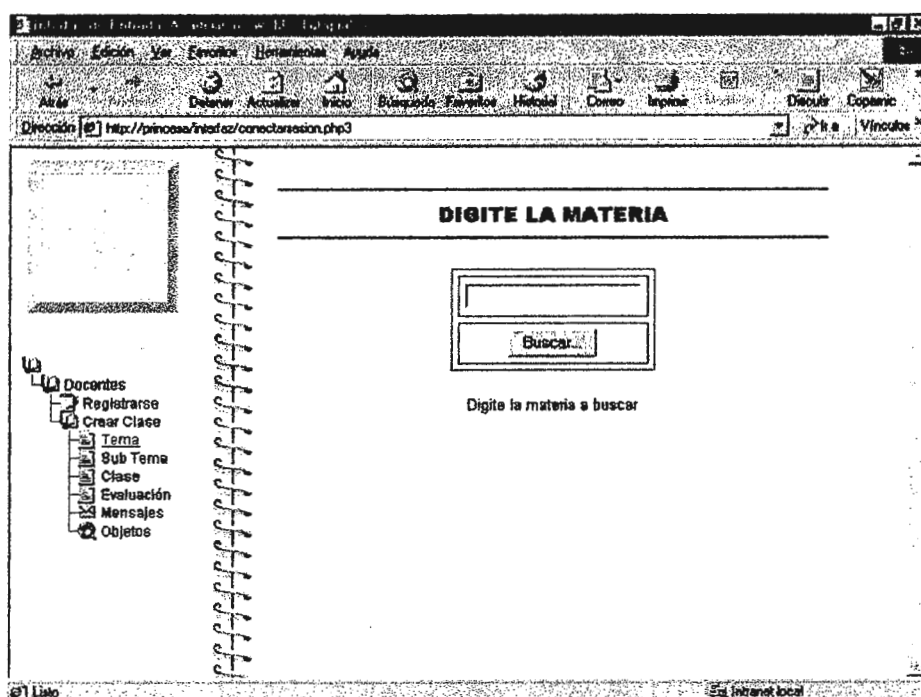


Pantalla Iniciar Sesión

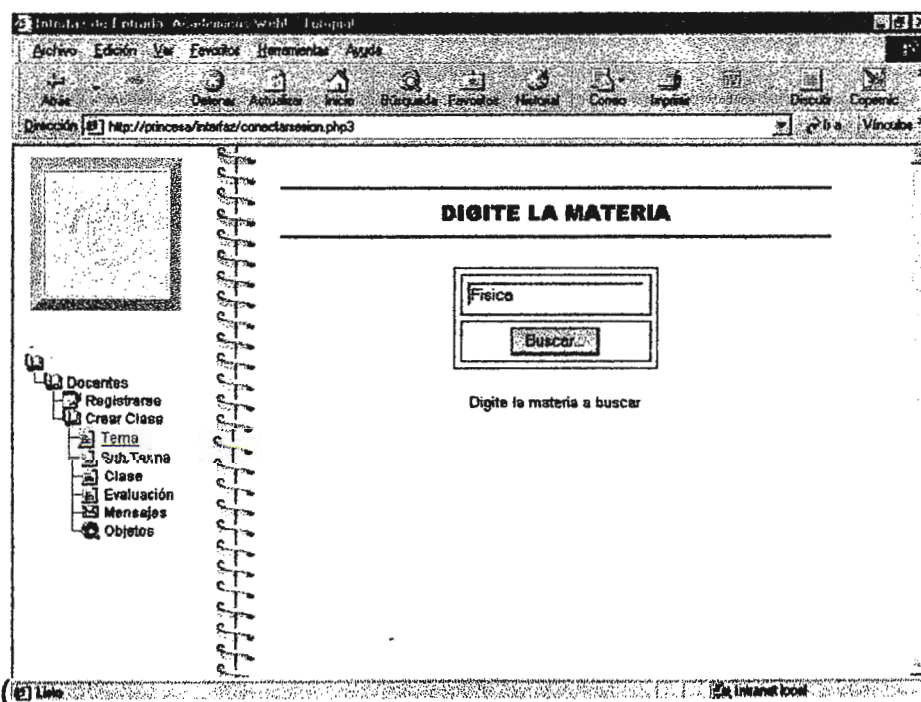
Cuando el usuario ya tiene asignado un usuario y una contraseña, esta listo para iniciar la sesión, después de digitarlos en la pantalla de bienvenida, se presenta la siguiente pantalla:



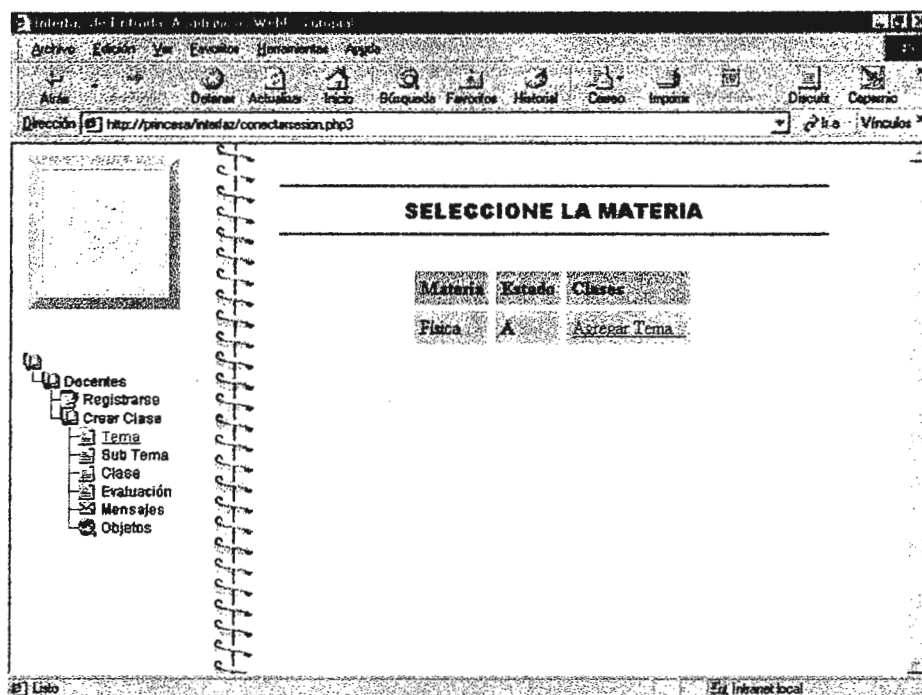
Al ingresar el docente debe de digitar el nombre de la materia dentro de la cual creara la clase, y dar clic al boton buscar.



Si el docente no digita nada en el campo para el nombre de la materia, se visualizaran todas las materias existentes, de lo contrario puede realizar la búsqueda con todos el nombre de la materia (física) o solamente con los primeros caracteres (fis)

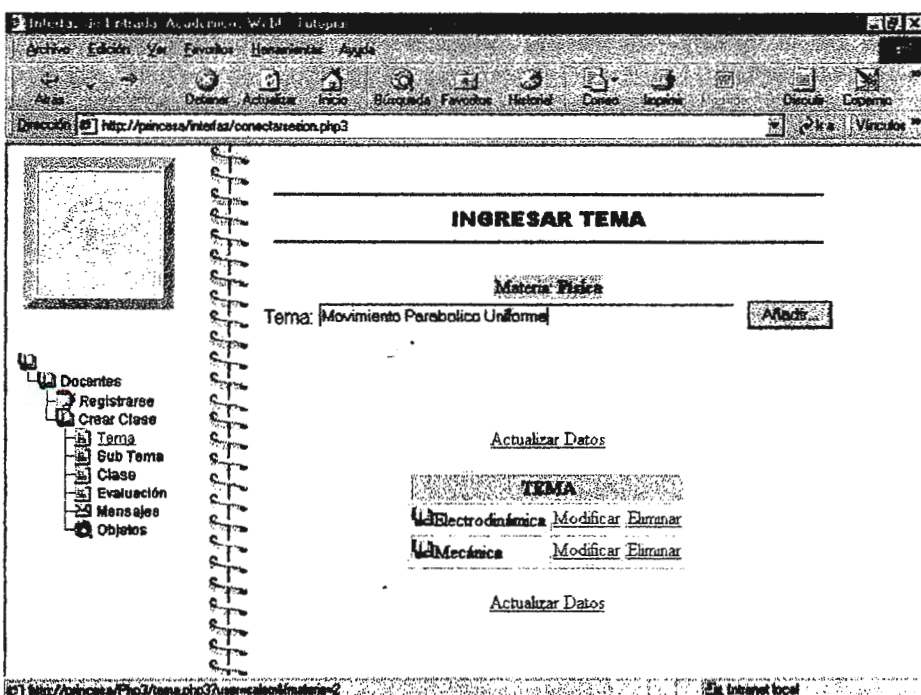
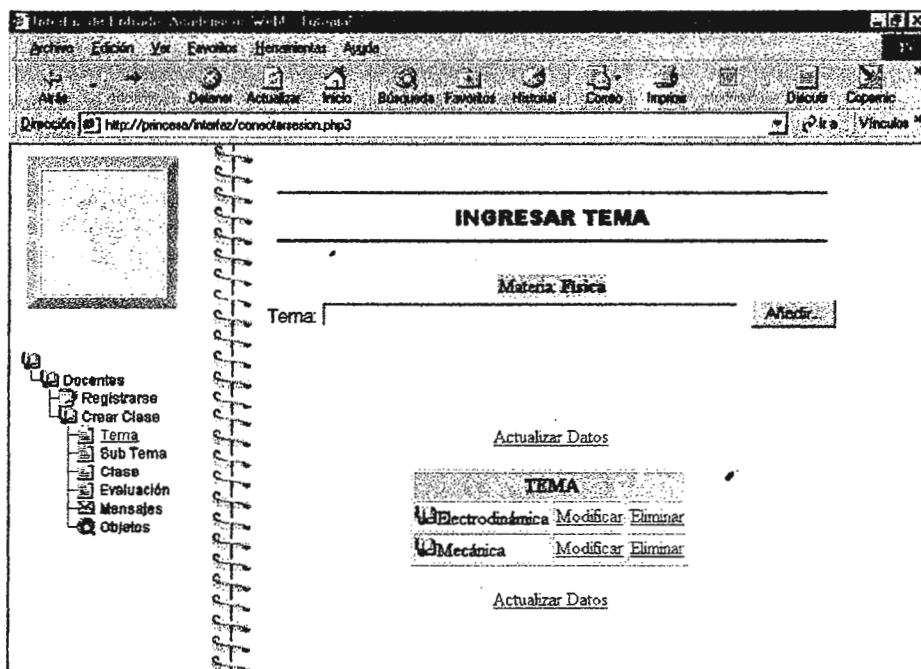


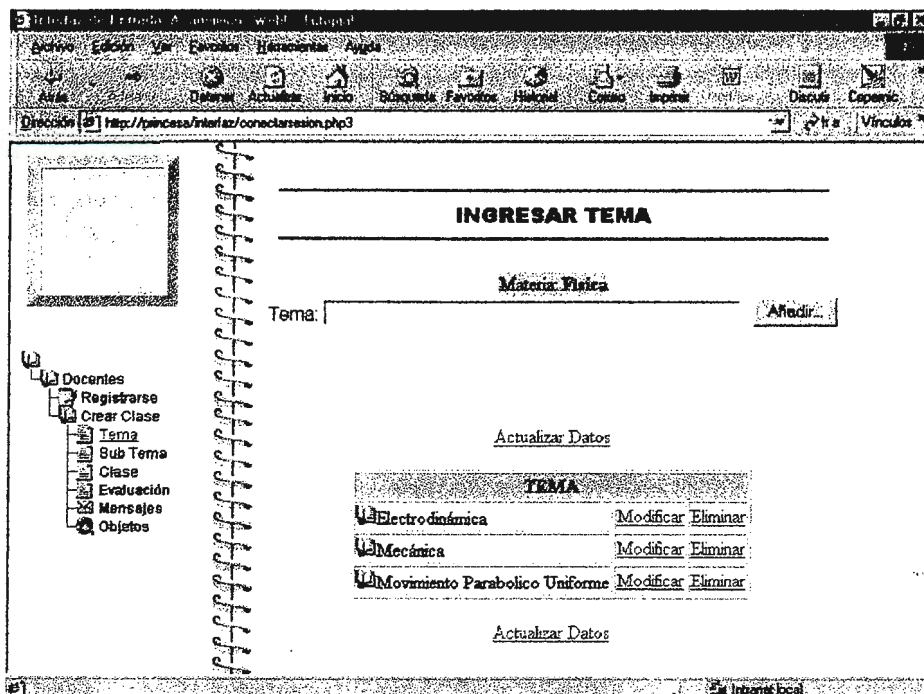
Después de seleccionar la materia, se presenta la opción de agregar tema.



Posteriormente se visualiza la pantalla donde aparece el nombre de la materia, y el campo donde se debe digitar el tema a desarrollar.

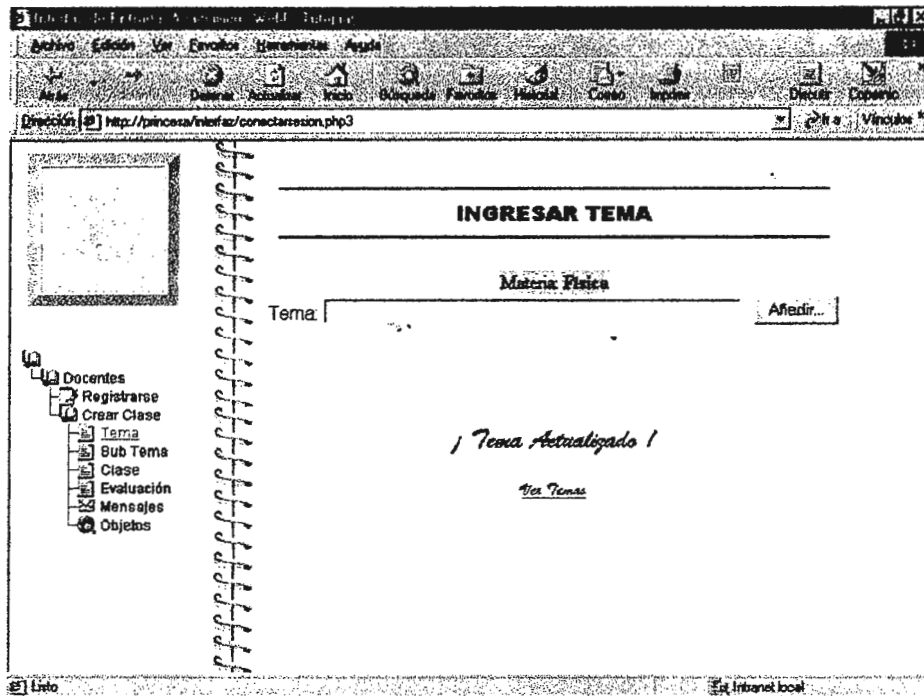
Después de digitado el tema se da clic a la opción añadir, y el tema se coloca en el área blanca de la pantalla, limpiándose el área de ingrese el nombre del tema, si el usuario da clic en la opción actualizar datos el tema se coloca dentro de la tabla donde ya posee la opción de modificar y eliminar, se pueden agregar tantos temas como el docente desee, la pantallas son las siguientes:



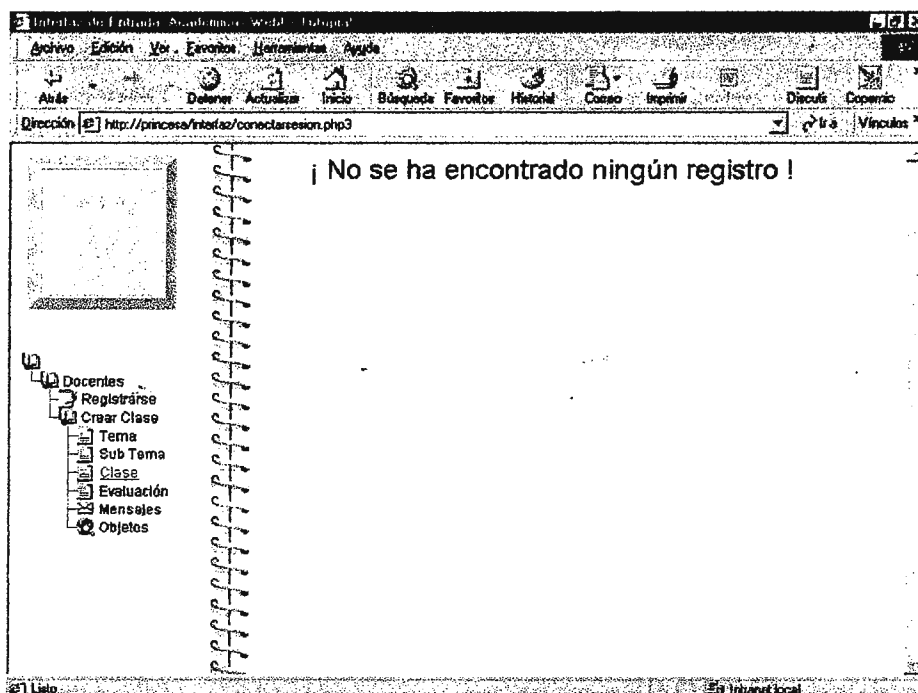


Si el usuario desea modificar el tema introducido, puede hacer uso de la opción modificar, que presenta la pantalla siguiente que contiene un campo que permite corregir el tema y un botón de actualizar.

La opción ver temas presenta la pantalla donde se visualizaban todos los temas ingresados. Si el tema se actualiza se visualiza un mensaje que indica que los datos han sido actualizados.



Si una materia, tiene un tema al cual no se le ha asignado ningún subtema se visualiza el mensaje "No se ha encontrado ningún registro", posteriormente se digita el subtema en el campo correspondiente, y se da clic en la opción actualizar datos y se ira formando una tabla que contendrá todos los subtemas ingresados por tema, con las opciones de modificar y eliminar.



Si el usuario desea realizar alguna modificación al subtema, utilizara la opción de modificar, y se presenta la pantalla que contiene el nombre de docente, materia, tema, y subtema. Así como también un área donde se actualiza el nombre del subtema, para que los cambios sean registrados se utiliza el botón de actualizar el cual valida las correcciones y presenta una pantalla donde se lee "Subtema actualizado"

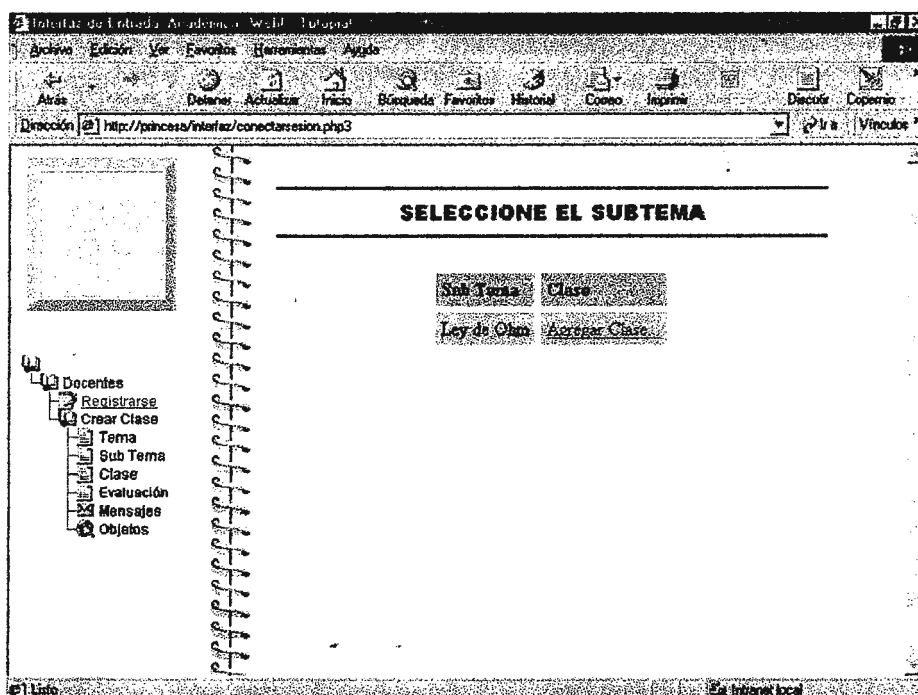
Si se da clic en ver subtemas se desplaza a la pantalla anterior donde ya se visualizan los cambios realizados.

La opción de eliminar subtema, presenta la pantalla donde se visualiza la información relacionada con cada subtema, si le damos clic al botón eliminar significara que borraremos el subtema del listado y se visualiza una pantalla con el

mensaje "Subtema eliminado", La opción ver subtemas permite visualizar el listado de subtemas actualizados.

Después de definir, la materia, el tema y el subtema, se procede al desarrollo de la clase, se empezara por seleccionar la materia a la cual pertenece el tema, luego entre la lista de temas y subtemas se selecciona el indicado.

La parte mas importante es cuando el usuario selecciona la opción de agregar su clase, donde se especificara el nombre de la clase, objetivo, duración en minutos, y dará clic en el botón actualizar, para que sea agregado al listado de clases desarrolladas para un tema específico.



Internet Explorer - http://princesa/interfaz/conectarasesion.php3

INGRESAR CLASE

Materia: Física Tema: Electrodinámica Tema: Ley de Ohm

Clase:

Nombre:

Objetivo:

Duración: 45

CLASES

La Ley de Ohm	Modificar	Eliminar	Crear Página	Escojer Clase
Primera Clase	Modificar	Eliminar	Crear Página	Escojer Clase

Docentes
 Registrarse
 Crear Clase
 Tema
 Sub Tema
 Clase
 Evaluación
 Mensajes
 Objetos

Si el docente desea modificar los datos de la clase con la opción modificar se presentara la siguiente pantalla, y el botón actualizar permite hacer validas dichas modificaciones.

Internet Explorer - http://princesa/interfaz/conectarasesion.php3

INGRESAR CLASE

Materia: Física Tema: Electrodinámica Tema: Ley de Ohm

Clase:

Nombre:

Objetivo:

Duración: 45

Actualice los datos de la clase:

Nombre:

Objetivo:

Duración:

Docentes
 Registrarse
 Crear Clase
 Tema
 Sub Tema
 Clase
 Evaluación
 Mensajes
 Objetos

Interfaz de Interacción Académica Web - Tutorial

Archivo Edición Ver Favoritos Herramientas Ayuda

Atás Detener Actualizar Inicio Búsqueda Favoritos Historial Correo Impedir Discutir Copiar

Dirección <http://princesa/interfaz/conexionresion.php3> Ir Vinculos

INGRESAR CLASE

Materia: Física Tema: Electrodinámica Tema: Ley de Ohm

Clase:

Nombre:

Objetivo:

Duración: 45

¡ Clase Actualizada !

[Ver Clases](#)

Intranet local

Opción: Eliminar Clase

Cuando los datos de la clase se han digitado, pero se desea eliminar la información correspondiente, se selecciona la opción de eliminar la clase, basta con darle clic al botón eliminar clase, para que se elimine y finalmente en la opción ver clases se visualiza el listado de las clases existentes y actualizadas.

Interfaz de Interacción Académica Web - Tutorial

Archivo Edición Ver Favoritos Herramientas Ayuda

Atás Detener Actualizar Inicio Búsqueda Favoritos Historial Correo Impedir Discutir Copiar

Dirección <http://princesa/interfaz/conexionresion.php3> Ir Vinculos

INGRESAR CLASE

Materia: Física Tema: Electrodinámica Tema: Ley de Ohm

Clase:

Nombre:

Objetivo:

Duración: 45

Materia: Física
Tema: Electrodinámica
Subtema: Ley de Ohm
Clase: Primera Clase

Desea eliminar la clase !

Intranet local

Pantalla de Generar Clase

Cuando el usuario desea generar la clase vera una pantalla como la siguiente, dentro de la cual debe de digitar la información necesaria para que su clase se estructure de la mejor manera, los componentes de esta pantalla son:

Encabezados: contiene tres tipos diferentes de formato para encabezados o títulos
G= grande m = mediano p = pequeño

Estilos: contiene las opciones de negrita, cursiva y subrayado

especial: contiene varios caracteres al usuario que lo puede insertar dentro del cuerpo de la clase

Letra: permite modificar el tipo de letra, el color y tamaño

Lista: Cuando se define información en formato de lista permite colocar etiquetas a cada elemento de la lista

Título de clase: se digita el título de la clase a desarrollar

Dirección de Email: El usuario digita su dirección de correo electrónico donde el alumno posteriormente enviara sus comentarios

Dirección URL: área donde el usuario puede digitar direcciones de pagina Web que tengan información referente al contenido de su clase, o información importante que sirva de apoyo al alumno.

Nombre de dirección: Alguna descripción que especifique el tipo de información que contiene el sitio recomendado

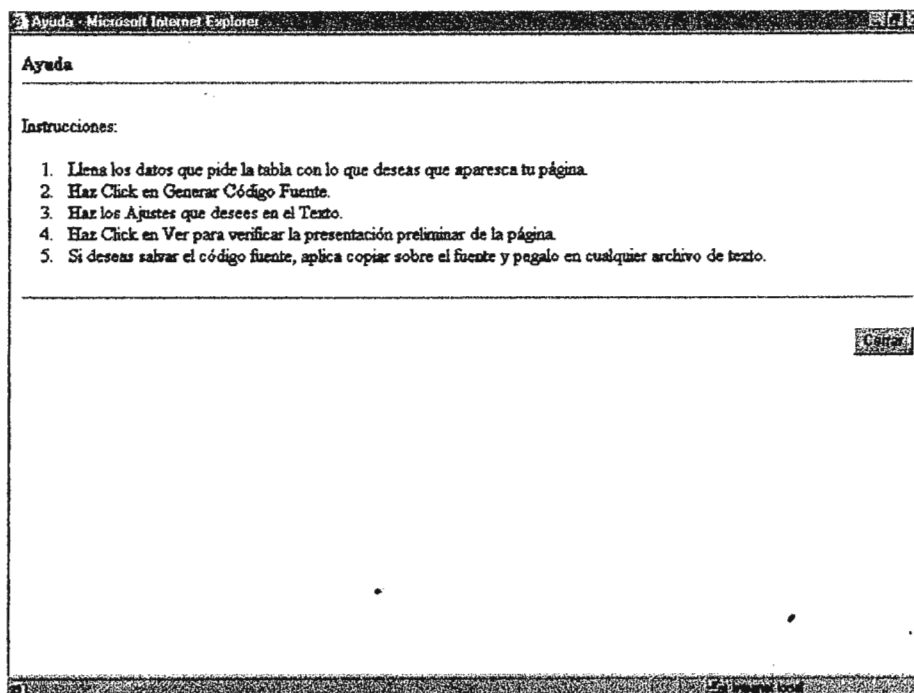
Ej.:

Dirección URL: <http://www.download.com>

Nombre de dirección: Software gratis

Cuerpo de la clase: Es toda el área donde el usuario digita el contenido de la clase, como introducción, definición de términos, formulas, etc.

Ayuda: Al darle clic en el botón ayuda se despliega una pantalla que contiene indicaciones importantes de uso de esta pantalla. Que es la siguiente



En las siguientes pantallas, el usuario ha comenzado a crear su clase, se visualiza una ventana adicional, si el usuario desea colocar un texto en negrita, cambiar tipo de letra etc. Debe darle clic al botón que tiene N, posteriormente se le cargara una ventana como la que se visualiza a continuación, en ella se digita el texto que queremos que aparezca en negrita, por ejemplo, en el área e cuerpo de la clase se vera el texto entre etiquetas Html, por las cuales el usuario no necesitara preocuparse, ya que al darle clic al botón generar pagina esto será interpretado y se visualizara la clase sin dichas etiquetas

Message de JavaScript
Ingreso el Texto a Editar en Negrita

Aceptar
Cancelar

Como Imprimir Modificar Descartar Copiar

214class=1

Generar Clase

ESTILOS

N C S

ESPECIAL

F P L

H - #

9 6

JUSTIFICADO

J C D

LETRA

Arial

Negro

2

LISTA

Bullet

Título Clase: Teoría Electrodinámica

Dirección e-Mail: valientemercade@efinmail.com

SITIOS INTERESANTES

Dirección(URL): **Nombre Dirección:**

<http://www.tutoriel.com> Tutor en Internet

<http://www.cibercursos.com> Cursos en Línea

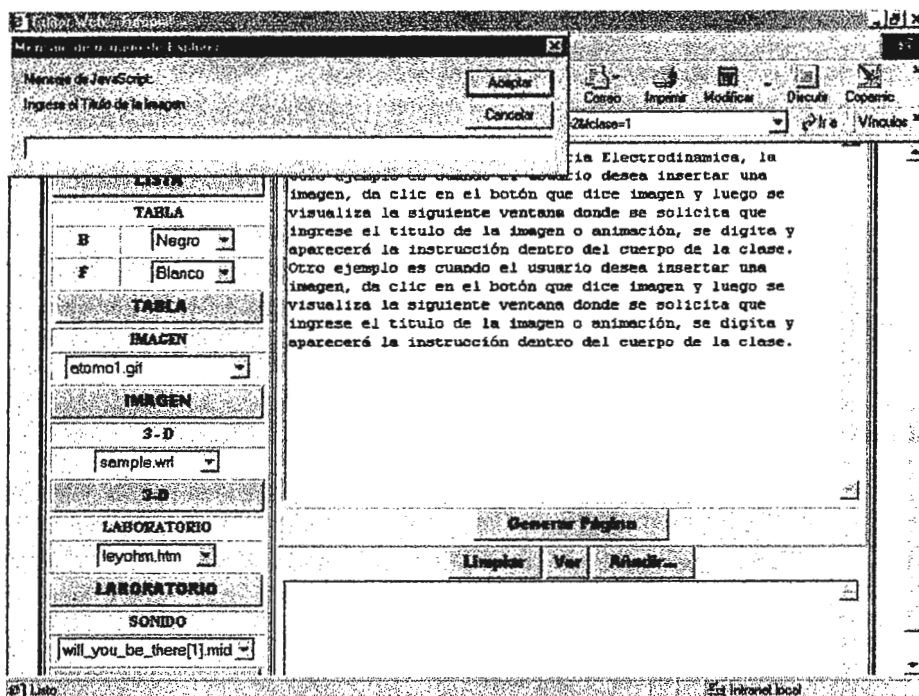
<http://www.download.com> Software gratis

CUERPO DE LA CLASE

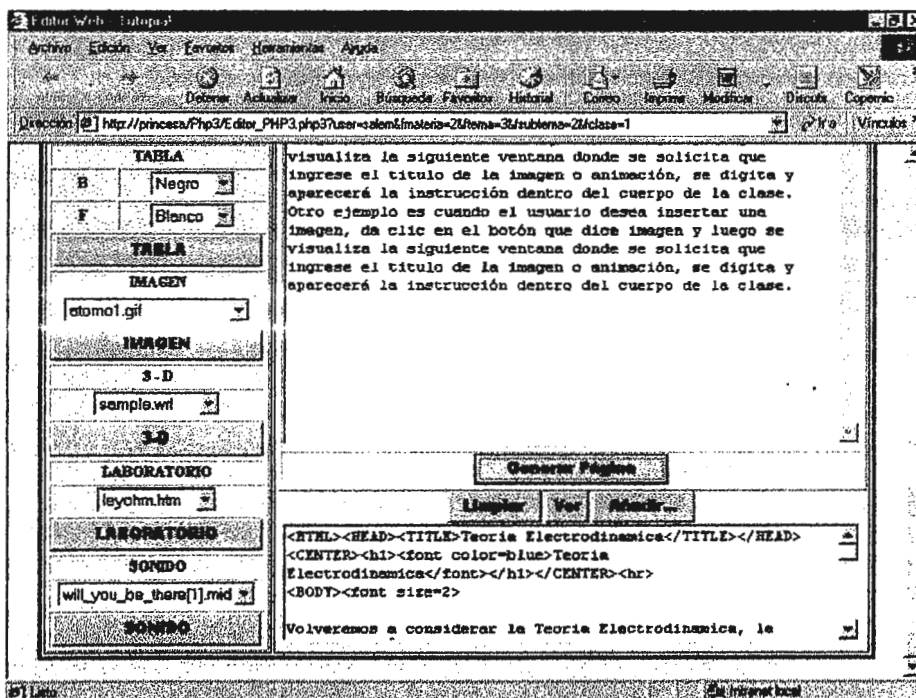
Volveremos a considerar la Teoría Electrodinámica, la

Libro Internet local

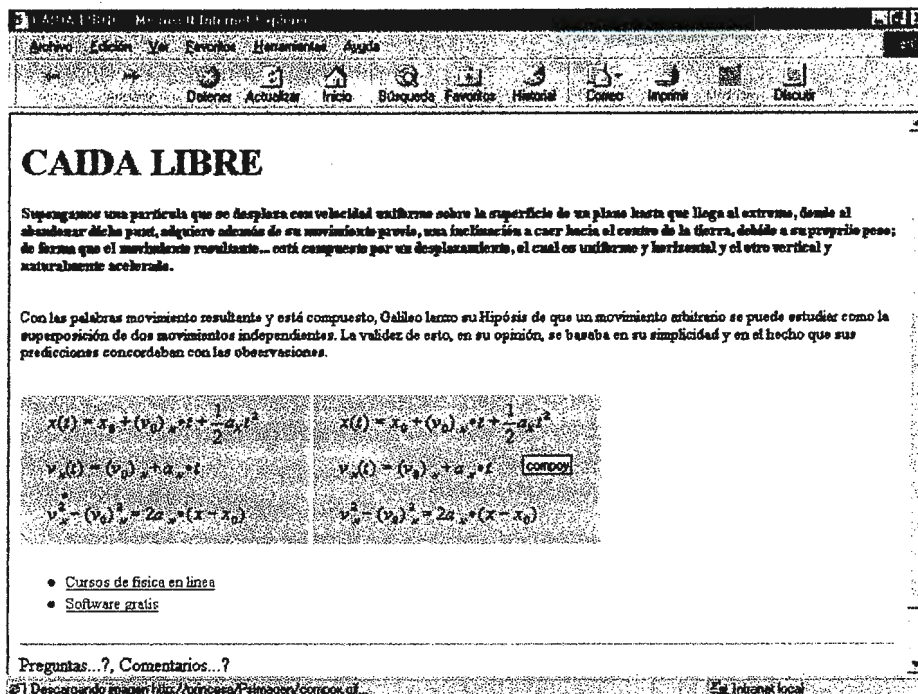
Otro ejemplo es cuando el usuario desea insertar una imagen, da clic en el botón que dice imagen y luego se visualiza la siguiente ventana donde se solicita que ingrese el título de la imagen o animación, se digita y aparecerá la instrucción dentro del cuerpo de la clase.



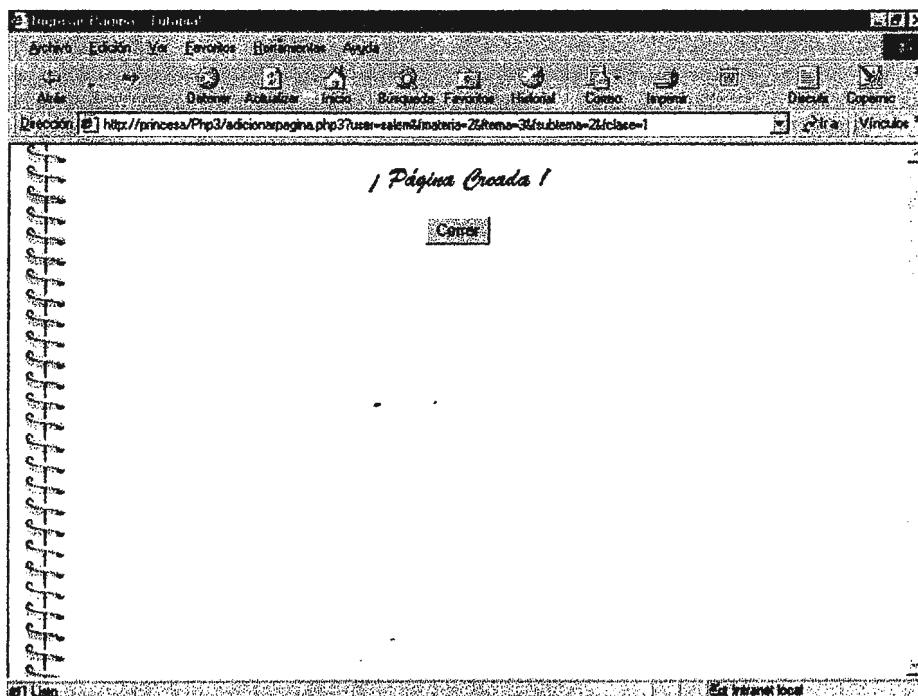
Cuando el docente termina de digitar su clase da clic en el botón generar pagina, y se presenta la pantalla siguiente:



Posteriormente el docente da clic en el botón generar pagina, y visualizara la clase ya terminada, con todas sus imágenes, laboratorios, links a direcciones en el web, etc. Un ejemplo de esta pantalla es el siguiente



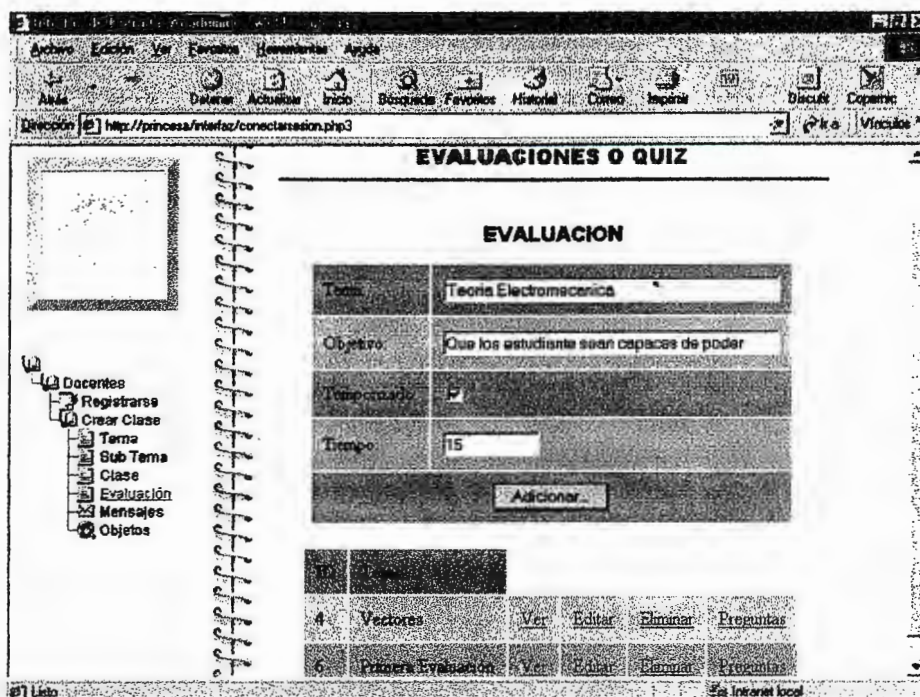
Para que el docente pueda colocar su clase a disposición, de los alumnos debe de darle un clic al botón añadir y muestra la pantalla siguiente:



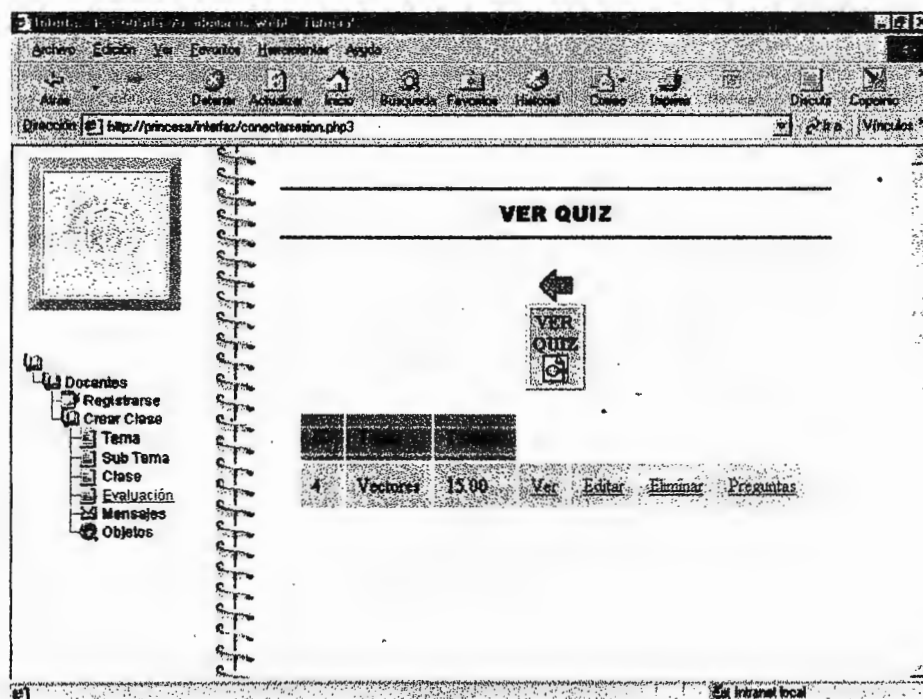
Dentro del menú principal, se encuentra la opción de hacer evaluación, al darle clic en esta opción, se visualiza una pantalla como el siguiente donde aparece el nombre de las materias, cada una con su opción Hacer Evaluación, al darle clic en hacer evaluación se presenta la pantalla que permite elaborar un test, el que será consultado por el alumno para medir sus conocimientos.



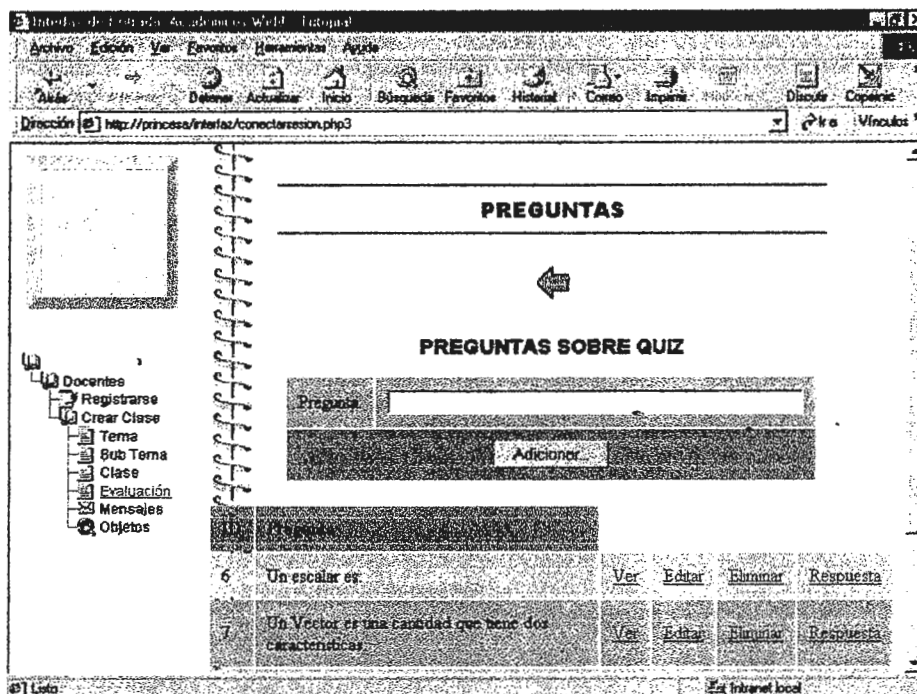
La opción de hacer evaluación despliega la siguiente pantalla, donde se especifica el tema, objetivo, evaluación, tiempo de duración de la prueba, y un checkbox el cual si esta activado indica que la prueba tiene tiempo de lo contrario el alumno puede tomarse el tiempo que desee, si esta tuviera tiempo se especifica de lo contrario no.



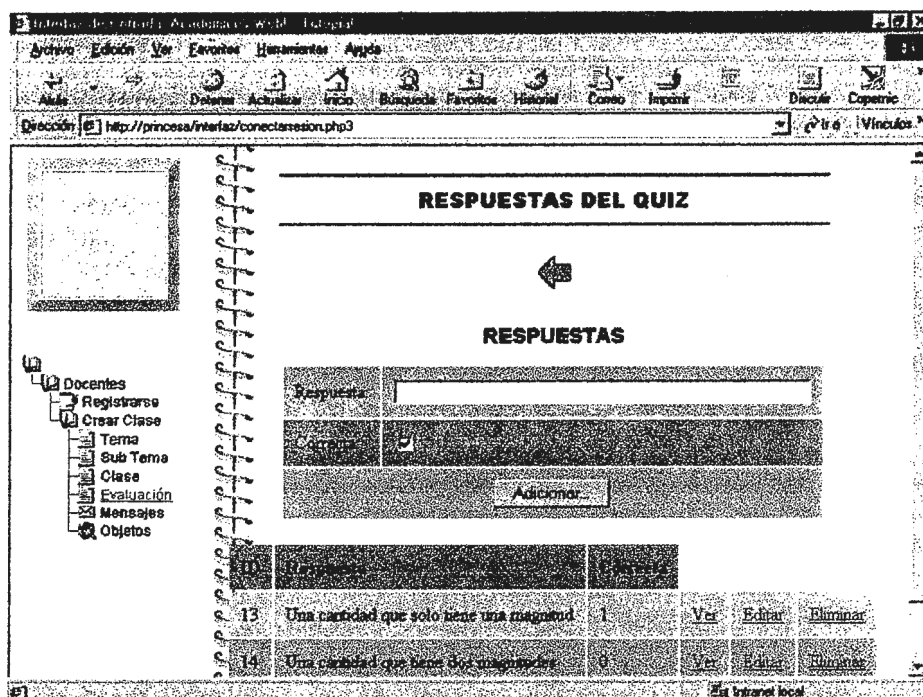
La siguiente pantalla muestra el tema, objetivo, tipo (la prueba posee tiempo = 1, no posee tiempo = 0), tiempo en segundos y las opciones de ver, editar, eliminar y preguntas



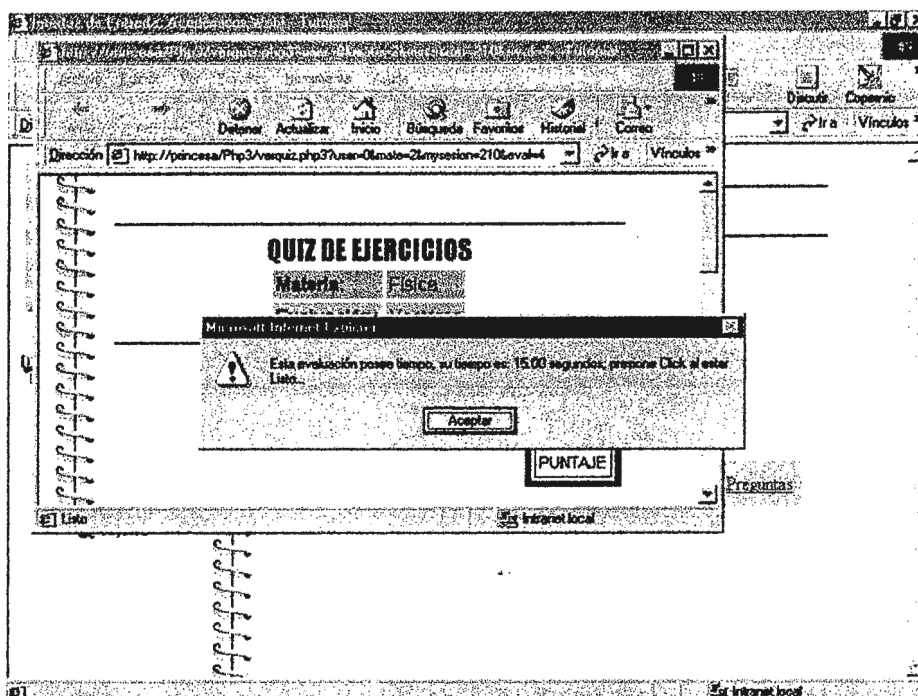
Cuando el docente elabora el test debe digitar las preguntas en la siguiente pantalla, el docente puede ingresar cuantas preguntas quiera, y cada pregunta puede tener varias respuestas, pero solo una es la correcta



Para que el docente digite las respuesta se hace desde la siguiente pantalla, donde a través del cual determina cual es la correcta, el cheque indica que es la correcta, si no esta activa la respuesta no es la correcta.

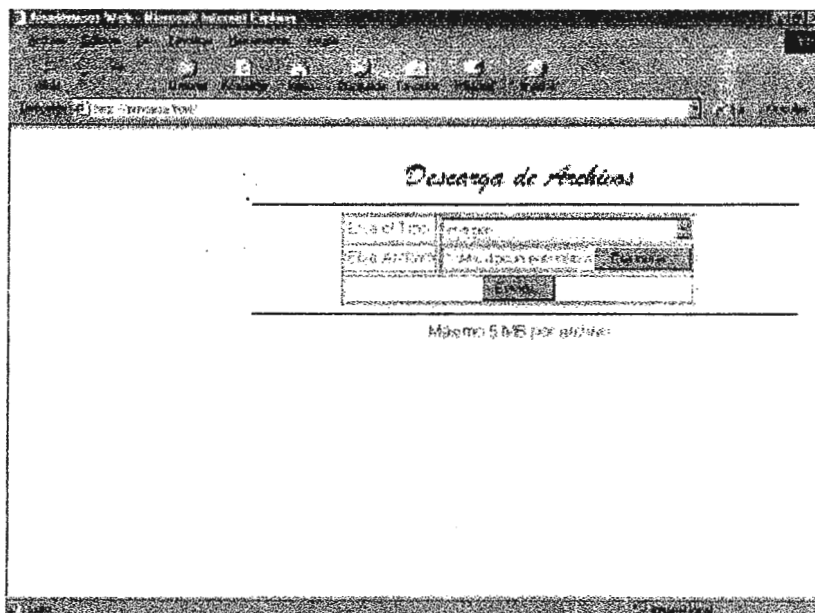


Por ejemplo el docente a elaborado un test con tiempo el cual al visualizarlo muestra una ventana donde se indica, al darle aceptar comienza a correr el tiempo, cuando el alumno hace clic en cada respuesta se despliega otra ventana que le indica si la respuesta es correcta o no.



Pantalla Objetos

Antes de crear la clase es aconsejable que el usuario realice la descarga de archivos, es decir copiara sus imágenes dentro de una archivo, que estará disponible cuando el cree su clase lo que le permitirá que dentro de la opción crear clase el pueda insertar las imágenes que desea, es aconsejable que realice este paso antes de crear su clase.



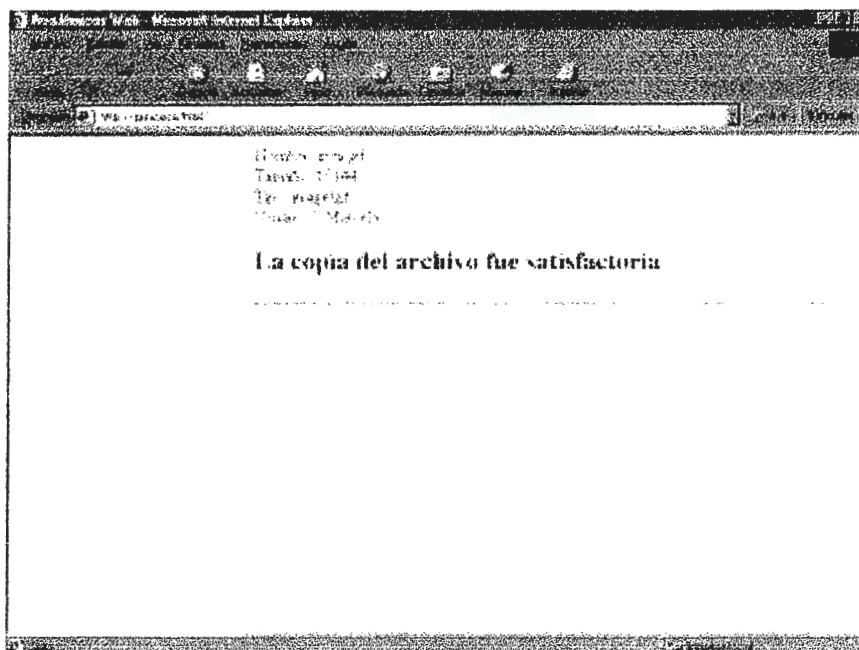
Descripción de campos:

Elija el tipo: Selecciona el tipo de archivo (imagen, vídeo)

Elija el archivo: Da clic en el botón examinar y el usuario selecciona la ubicación donde se encuentra el archivo cuyo tamaño máximo es de 5 Mb.

Al darle clic al botón enviar el archivo es copiado a un directorio específico, lo que indica que la imagen estará disponible en el desarrollo de la clase.

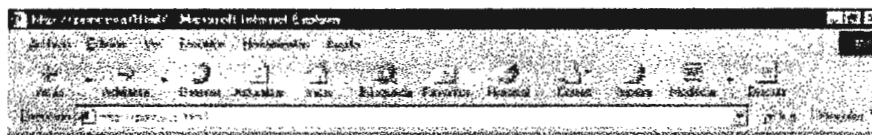
Pantalla de mensaje de envío de archivo



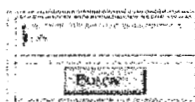
OPCIONES DEL MENU ALUMNOS

Pantalla Consultar Clase

Desde esta pantalla el alumno visualiza cada una de las clases creadas por un docente especifico, al darle clic en esta opción se presenta la pantalla donde se busca al docente



Ingresa el nombre del docente
a buscar



29



Seleccione el diccionario

Alfonso Aparicio Sánchez
María Isabel Sánchez

luego se selecciona el tema, subtema, clase, la pagina, y se selecciona la opción consultar que presenta el resultado final de la clase.



CAIDA LIBRE

Supongamos una partícula que se desplaza con velocidad constante sobre la superficie de un plano hasta que llega al extremo, donde al abandonar dicho punto adquiere velocidad de movimiento por una inclinación a caer hacia el centro de la tierra, debido a su propia peso; de forma que el movimiento resultante... está compuesto por un desplazamiento, el cual es uniforme y horizontal) y el otro vertical y uniformemente acelerado.

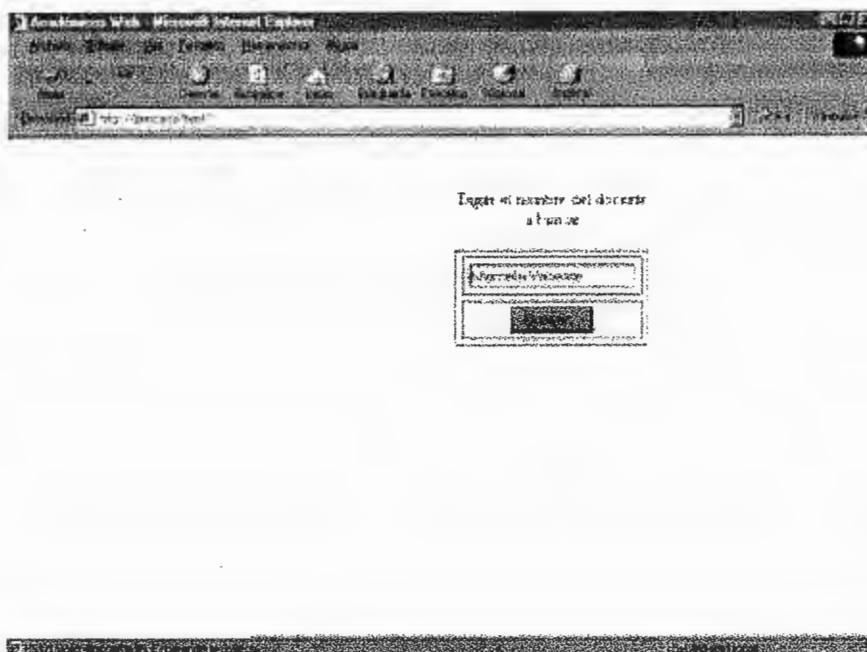
Con los puntos de movimiento conocidos y esta composición, obtenemos la Hipótesis de que un movimiento uniforme se puede considerar como la superposición de dos movimientos independientes, uno vertical de caída libre y otro horizontal de velocidad constante y se obtiene que sus ecuaciones de movimiento son las siguientes:

$$\begin{aligned} x(t) &= x_0 + (v_x)_0 t + \frac{1}{2} a_x t^2 \\ x(t) &= (v_x)_0 t \\ v_x^2 &= (v_x)_0^2 + 2a_x(x - x_0) \end{aligned} \quad \begin{aligned} y(t) &= y_0 + (v_y)_0 t + \frac{1}{2} a_y t^2 \\ y(t) &= (v_y)_0 t + \frac{1}{2} a_y t^2 \\ v_y^2 &= (v_y)_0^2 + 2a_y(y - y_0) \end{aligned}$$

- Ecuación de Caída Libre
- Ecuación de Velocidad

Pantalla Evaluaciones

El alumno tiene la opción de realizar exámenes desarrollados por cada docente, donde selecciona el nombre del docente, y cada uno tiene asociado una serie de evaluaciones a realizar de acuerdo a la materia y el tema desarrollado.



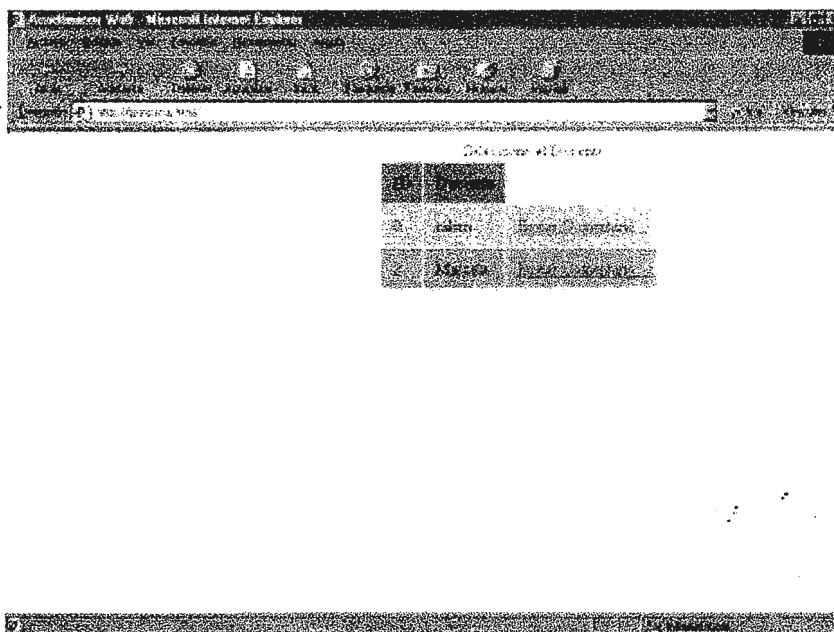
Pantalla Enviar comentarios

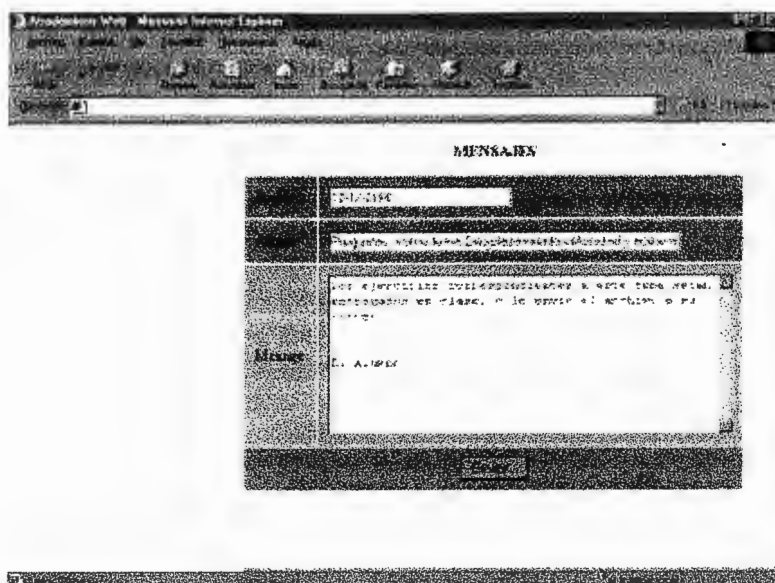
Si el alumno tiene alguna observación, o duda sobre un tema específico debe utilizar esta opción, por medio de la envia un mensaje al docente.

Los pasos son:

- seleccionar docente
- dar clic en la opción enviar comentario

Y se presenta la siguiente pantalla





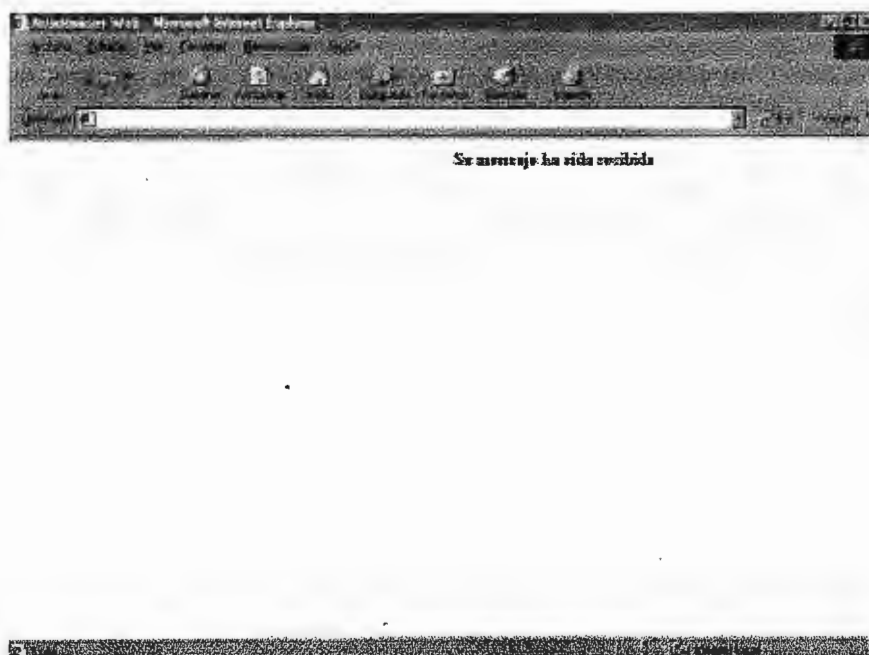
Descripción de campos:

Fecha: Fecha en la cual se elabora el mensaje.

Asunto: Breve descripción del contenido del mensaje.

Mensaje: Area donde se digita el contenido del mensaje.

Cuando el alumno da clic en la opción enviar comentario se presenta la siguiente pantalla.



Manual Técnico

I.1 Lista por grupos de programas

A continuación se presenta un listado que contiene el nombre de cada uno de los programas utilizados en el desarrollo de la aplicación, los cuales pueden ser accedidos desde cualquier Editor de Texto (Word, Front Page, Notepad, WordPad, Homesite, entre otros) para futuras mejoras.

Ver Tabla 1 y Tabla 1.1

I.2 Estructura de Datos a utilizar para levantar aplicación

```
# phpMyAdmin MySQL-Dump
# http://phpwizard.net/phpMyAdmin/
#
# Host: princesa Database : Asistente
```

```
# -----
#
# Table structure for table 'ALUMNOS'
#
```

```
DROP TABLE IF EXISTS ALUMNOS;
CREATE TABLE ALUMNOS (
  Alumno_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Nombre varchar(60) NOT NULL,
  Apellido varchar(60) NOT NULL,
  Email varchar(41) NOT NULL,
  Usuario varchar(30) NOT NULL,
  Contraseña varchar(30) NOT NULL,
  Estatus int(1) DEFAULT '0' NOT NULL,
  PRIMARY KEY (Alumno_ID)
);
```

```
#
# Dumping data for table 'ALUMNOS'
#
```

```
INSERT INTO ALUMNOS VALUES( '0', 'Salem Emilio', 'salem', 'salem@salem.com', 'ssalem',
'4e7e92b063c6d8a9', '1');
INSERT INTO ALUMNOS VALUES( '1', 'Alumno', 'Alumno', 'Alumno@mail.com', 'alumno',
'1b25731233d5051d', '1');
```

```
# -----
#
# Table structure for table 'BIBLIOTECAENC'
#
```

```
DROP TABLE IF EXISTS BIBLIOTECAENC;
CREATE TABLE BIBLIOTECAENC (
```

```

Archivo_ID int(11) DEFAULT '0' NOT NULL,
ruta varchar(100),
descripcion varchar(50),
tipo char(1),
Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
PRIMARY KEY (Archivo_ID, Docente_ID)
);

```

```

#
# Dumping data for table 'BIBLIOTECAENC'
#

```

```

INSERT INTO BIBLIOTECAENC VALUES( '1', '../Pslmagen/atomo1.gif', 'atomo1.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '2', '../Pslmagen/Pabajo.gif', 'Pabajo.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '3', '../Pslmagen/transp.gif', 'transp.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '4', '../Pslinteractividad/leyohm.htm', 'leyohm.htm', 'A', '0');
INSERT INTO BIBLIOTECAENC VALUES( '5', '../Pslmagen/btnBack.gif', 'btnBack.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '6', '../Pslmagen/btnForward.gif', 'btnForward.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '7', '../Pslmagen/btnHome.gif', 'btnHome.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '8', '../Pslmagen/distint_verde.gif', 'distint_verde.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '9', '../Pslmagen/vect_graf_06.gif', 'vect_graf_06.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '10', '../Pslmagen/vect_graf_07.gif', 'vect_graf_07.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '11', '../Pslmagen/vect_graf_08.gif', 'vect_graf_08.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '12', '../Pslmagen/vect_graf_09.gif', 'vect_graf_09.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '13', '../Pslmagen/vect_graf_10.gif', 'vect_graf_10.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '14', '../Pslmagen/vect_graf_11.gif', 'vect_graf_11.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '15', '../Pslmagen/vect_graf_12.gif', 'vect_graf_12.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '16', '../Pslmagen/vect_graf_13.gif', 'vect_graf_13.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '17', '../Pslmagen/vect_graf_14.gif', 'vect_graf_14.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '18', '../Pslmagen/vect_graf_15.gif', 'vect_graf_15.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '19', '../Pslmagen/vect_graf_16.gif', 'vect_graf_16.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '20', '../Pslmagen/vect_graf_16.gif', 'vect_graf_16.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '21', '../Pslmagen/vector_A_azul.gif', 'vector_A_azul.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '22', '../Pslmagen/vector_A_blanco.gif', 'vector_A_blanco.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '23', '../Pslmagen/vector_A_marron.gif', 'vector_A_marron.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '24', '../Pslmagen/vector_A_verde.gif', 'vector_A_verde.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '25', '../Pslmagen/vector_AB_azul.gif', 'vector_AB_azul.gif', 'I', '0');

```



```

INSERT INTO BIBLIOTECAENC VALUES( '26', '..\PsImagen/vector_AB_marron.gif',
'vector_AB_marron.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '27', '..\PsImagen/vector_Am_marron.gif',
'vector_Am_marron.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '28', '..\PsImagen/vector_Amu_marron.gif',
'vector_Amu_marron.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '29', '..\PsImagen/vector_Amu_verde.gif',
'vector_Amu_verde.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '30', '..\PsImagen/vector_B_azul.gif', 'vector_B_azul.gif',
'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '31', '..\PsImagen/vector_B_blanco.gif',
'vector_B_blanco.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '32', '..\PsImagen/vector_B_marron.gif',
'vector_B_marron.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '33', '..\PsImagen/vector_B_marron.gif',
'vector_B_marron.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '34', '..\PsImagen/vector_C_blanco.gif',
'vector_C_blanco.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '35', '..\PsImagen/vector_C_marron.gif',
'vector_C_marron.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '36', '..\PsImagen/vector_CD_marron.gif',
'vector_CD_marron.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '37', '..\PsImagen/vector_i_azul.gif', 'vector_i_azul.gif',
'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '38', '..\PsImagen/vector_i_azul.gif', 'vector_i_azul.gif',
'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '39', '..\PsImagen/vector_i_verde.gif',
'vector_i_verde.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '40', '..\PsImagen/vector_j_azul.gif', 'vector_j_azul.gif',
'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '41', '..\PsImagen/vector_j_verde.gif',
'vector_j_verde.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '42', '..\PsImagen/vector_k_azul.gif', 'vector_k_azul.gif',
'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '43', '..\PsImagen/vector_k_verde.gif',
'vector_k_verde.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '44', '..\PsImagen/vector_O_blanco.gif',
'vector_O_blanco.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '45', '..\PsImagen/vector_O_verde.gif',
'vector_O_verde.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '46', '..\PsInteractividad/resultesp.htm', 'resultesp.htm',
'A', '0');
INSERT INTO BIBLIOTECAENC VALUES( '47', '..\PsSonido\will_you_be_there[1].mid',
'will_you_be_there[1].mid', 'S', '0');
INSERT INTO BIBLIOTECAENC VALUES( '48', '..\PsSonido\thriller.mid', 'thriller.mid', 'S', '0');
INSERT INTO BIBLIOTECAENC VALUES( '49', '..\PsVrml/sample.wrl', 'sample.wrl', 'V', '0');
INSERT INTO BIBLIOTECAENC VALUES( '50', '..\PsVrml/sample.wrl', 'sample.wrl', 'V', '0');
INSERT INTO BIBLIOTECAENC VALUES( '51', '..\PsVrml/example8.wrl', 'example8.wrl', 'V', '0');
INSERT INTO BIBLIOTECAENC VALUES( '52', '..\PsVrml/example10.wrl', 'example10.wrl', 'V', '0');
INSERT INTO BIBLIOTECAENC VALUES( '53', '..\PsVrml/example7.wrl', 'example7.wrl', 'V', '0');
INSERT INTO BIBLIOTECAENC VALUES( '59', '..\PsImagen\Carbn_01.mid', 'Carbn_01.mid', 'I',
'0');
INSERT INTO BIBLIOTECAENC VALUES( '55', '..\PsInteractividad\fahr1.gif', 'fahr1.gif', 'A', '0');
INSERT INTO BIBLIOTECAENC VALUES( '56', '..\PsImagen\fahr2.gif', 'fahr2.gif', 'I', '0');
INSERT INTO BIBLIOTECAENC VALUES( '57', '..\PsInteractividad\n2ley.htm', 'n2ley.htm', 'A', '0');
INSERT INTO BIBLIOTECAENC VALUES( '58', '..\PsInteractividad\fahr2.gif', 'fahr2.gif', 'A', '0');

```



```
# -----
#
# Table structure for table 'CLASES'
#
```

```
DROP TABLE IF EXISTS CLASES;
CREATE TABLE CLASES (
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Materia_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Tema_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Subtema_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Clase_ID decimal(10,0) DEFAULT '0' NOT NULL,
  nombrecase varchar(50) NOT NULL,
  objetivocase varchar(50),
  fechacreacion date DEFAULT '0000-00-00' NOT NULL,
  fechapresentacion date DEFAULT '0000-00-00' NOT NULL,
  duracion decimal(15,0),
  tamañoocase decimal(15,0),
  PRIMARY KEY (Docente_ID, Materia_ID, Tema_ID, Subtema_ID, Clase_ID)
);
```

```
#
# Dumping data for table 'CLASES'
#
```

```
INSERT INTO CLASES VALUES( '0', '1', '0', '0', '0', 'Primera Clase', 'Conocer la Primera Clase de
Mate', '2000-09-12', '2000-09-12', '45', '5');
INSERT INTO CLASES VALUES( '0', '2', '3', '2', '1', 'La Ley de Ohm', 'Conocer la ley de Ohm',
'2000-09-16', '2000-09-16', '45', '5');
INSERT INTO CLASES VALUES( '0', '2', '2', '3', '2', 'Vectores', 'Conocer la definición de Vector',
'2000-09-16', '2000-09-16', '45', '5');
INSERT INTO CLASES VALUES( '0', '3', '4', '4', '3', 'clase1', 'lakfjlskfd', '2000-09-16', '2000-09-16',
'45', '5');
INSERT INTO CLASES VALUES( '0', '1', '0', '5', '4', 'matre1', 'lñdfñgiklwerjliwekj', '2000-09-17',
'2000-09-17', '45', '5');
```

```
# -----
#
# Table structure for table 'DOCENTES'
#
```

```
DROP TABLE IF EXISTS DOCENTES;
CREATE TABLE DOCENTES (
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
  nombres varchar(30) NOT NULL,
  apellidos varchar(30),
  email varchar(50),
  estadodocente char(1) NOT NULL,
  nombreusuario varchar(16) NOT NULL,
  contraseñausuario varchar(16) NOT NULL,
  profesion varchar(30),
  cargo varchar(30),
  pregunta varchar(40),
  respuesta varchar(40),
  direccion varchar(120),
```

```
telefono varchar(25),
fax varchar(25),
fechaingreso date DEFAULT '0000-00-00',
PRIMARY KEY (Docente_ID)
```

```
);
```

```
#
```

```
# Dumping data for table 'DOCENTES'
```

```
#
```

```
INSERT INTO DOCENTES VALUES( '0', 'Salem', 'Salem', 'Salem_Zablah@hotmail.com', 'N',
'salem', '36102a9015b406e9', 'Ing. Computacion', 'Analista de Sistemas', '¿ Su cumpleaños ?',
'24febrero', 'Residencial el Volcan', '262-1303', '262-1303', '2000-05-05');
```

```
INSERT INTO DOCENTES VALUES( '3', 'Carlos', 'Bran', 'cbran@citt.cdb.edu.sv', 'N', 'locotus',
'3b920bf545f6ea6d', 'Ing. Electrica', 'PTC', '¿ Su cumpleaños ?', '14 de febrero', 'sdjhaskj', '123231',
'712368127', '2000-09-17');
```

```
INSERT INTO DOCENTES VALUES( '2', 'hgfhgfh', '', '', 'N', '', '', 'Ing. Computacion', '', '¿ Su
cumpleaños ?', '', '', '', '2000-09-17');
```

```
INSERT INTO DOCENTES VALUES( '1', 'Docente', 'Docente', 'Docente@mail.com', 'N', 'docente',
'7bda5c500b057026', 'Ing. Computacion', 'd', '¿ Su cumpleaños ?', '2', 'd', '2', '2', '2000-09-15');
```

```
# -----
```

```
#
```

```
# Table structure for table 'EVALUACIONENC'
```

```
#
```

```
DROP TABLE IF EXISTS EVALUACIONENC;
```

```
CREATE TABLE EVALUACIONENC (
```

```
  Materia_ID decimal(10,0) DEFAULT '0' NOT NULL,
```

```
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
```

```
  Evalua_ID decimal(10,0) DEFAULT '0' NOT NULL,
```

```
  Tema varchar(60) NOT NULL,
```

```
  Objetivo varchar(120) NOT NULL,
```

```
  Tipo int(5) DEFAULT '0' NOT NULL,
```

```
  Tiempo decimal(12,2) DEFAULT '0.00' NOT NULL,
```

```
  PRIMARY KEY (Materia_ID, Docente_ID, Evalua_ID)
```

```
);
```

```
#
```

```
# Dumping data for table 'EVALUACIONENC'
```

```
#
```

```
INSERT INTO EVALUACIONENC VALUES( '1', '0', '1', 'Primera Evaluación', 'Medir la capacidad
del Alumno', '1', '20.00');
```

```
INSERT INTO EVALUACIONENC VALUES( '1', '0', '2', 'Segunda Evaluación', 'Medir la capacidad
del Alumno', '1', '20.00');
```

```
INSERT INTO EVALUACIONENC VALUES( '1', '0', '3', 'Tercera Evaluación', 'Medir el
Conocimiento', '0', '0.00');
```

```
INSERT INTO EVALUACIONENC VALUES( '2', '0', '4', 'Vectores', 'Evaluar los conocimientos en
vectores', '1', '15.00');
```

```
INSERT INTO EVALUACIONENC VALUES( '1', '0', '5', 'unosas', 'jfhjfhjdjfhjdjfd', '1', '15.00');
```

```
# -----
```

```
#
```

```
# Table structure for table 'MATERIAS'
```

```
#
```

```
DROP TABLE IF EXISTS MATERIAS;
CREATE TABLE MATERIAS (
  Materia_ID int(10) DEFAULT '0' NOT NULL,
  nombremateria varchar(40) NOT NULL,
  estadomateria char(1) NOT NULL,
  PRIMARY KEY (Materia_ID),
  KEY Materia_ID (Materia_ID)
);

#
# Dumping data for table 'MATERIAS'
#

INSERT INTO MATERIAS VALUES( '1', 'Matematica', 'A');
INSERT INTO MATERIAS VALUES( '2', 'Física', 'A');
INSERT INTO MATERIAS VALUES( '3', 'Estadística', 'A');
INSERT INTO MATERIAS VALUES( '4', 'Naturales', 'A');
```

```
# -----
#
# Table structure for table 'MATERIASDOCENTES'
#
```

```
DROP TABLE IF EXISTS MATERIASDOCENTES;
CREATE TABLE MATERIASDOCENTES (
  Materia_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
  ciclo varchar(15) NOT NULL,
  PRIMARY KEY (Materia_ID, Docente_ID)
);
```

```
#
# Dumping data for table 'MATERIASDOCENTES'
#
```

```
# -----
#
# Table structure for table 'MENSAJES'
#
```

```
DROP TABLE IF EXISTS MENSAJES;
CREATE TABLE MENSAJES (
  Mensaje_ID int(11) DEFAULT '0' NOT NULL,
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
  fecha date DEFAULT '0000-00-00' NOT NULL,
  mensaje varchar(255) NOT NULL,
  estatus char(1) NOT NULL,
  Asunto varchar(50) NOT NULL,
  PRIMARY KEY (Mensaje_ID)
);
```

```
#
# Dumping data for table 'MENSAJES'
#
```

```

<STYLE type=text/css>A:active {
    COLOR: #ff0000; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
A:hover {
    COLOR: #009999; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
A:link {
    COLOR: #3333ff; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
A:visited {
    COLOR: #6666ff; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
.destacado {
    COLOR: #000066; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
.superdestacado {
    COLOR: #990000; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
.primer_a_letra {
    COLOR: #990000; FONT-FAMILY: \\"Times New Roman\\", Times, serif; FONT-SIZE: 18pt;
    FONT-WEIGHT: bold
}
.letra_destacada {
    COLOR: #ff3333; FONT-WEIGHT: bold
}
</STYLE>

```

```

<META content=\\"none, default\\" name=\\"Microsoft Theme\\">
<META content=\\"none, default\\" name=\\"Microsoft Border\\">
<META content=\\"MSHTML 5.00.2314.1000\\" name=GENERATOR></HEAD>
<BODY aLink=#cc0000 bgColor=#009eea link=#ff0000 text=#000000 vLink=#000000>
<P class=destacado><A name=Arriba></A></P>
<DIV align=center>
<CENTER>
<TABLE border=0 cellPadding=0 cellSpacing=0 width=\\"100%\\">
  <TBODY>
    <TR bgColor=#000000 vAlign=center>
      <TD align=middle vAlign=center width=\\"2%\\"><IMG height=30 hspace=10
        src=\\"../Pslmagen/atomo1.gif\\" width=30></TD>
      <TD align=middle vAlign=center
        width=\\"94%\\"><!-- #BeginEditable \\"TEMA\\" --><FONT color=#ffffff
        face=\\"Courier New, Courier, mono\\" size=5>Magnitudes eléctricas</FONT><!-- #EndEditable -->
    </TD>
      <TD align=middle vAlign=center width=\\"2%\\"><IMG height=30 hspace=10
        src=\\"../Pslmagen/atomo1.gif\\"
        width=30></TD></TR></TBODY></TABLE></CENTER></DIV>
<DIV align=center>
<CENTER>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width=\\"100%\\">

```

```

<TBODY>
<TR>
  <TD align=middle colSpan=3 vAlign=center><IMG height=8
    src=\"../Pslmagen/transp.gif\"
width=8></TD></TR>
<TR bgColor=#33ffff>
  <TD align=middle vAlign=baseline width=20> </TD>
  <TD align=left width=100%><!-- #BeginEditable \"APARTADO1\" --><FONT
    color=#000000 face=\"Courier New, Courier, mono\" size=4>Ley de Ohm</FONT><!--
#EndEditable --></TD>
  <TD align=middle vAlign=top width=20><IMG
    alt=Abajo border=0 height=20 hspace=20
    src=\"../Pslmagen/Pabajo.gif\"
width=25></TD></TR></TBODY></TABLE></CENTER></DIV>
<TABLE align=center bgColor=#ffffff border=0 cellPadding=20 cellSpacing=0
width=100%>
  <TBODY>
  <TR bgColor=#ffffff>
    <TD><!-- #BeginEditable \"TEXT0\" -->
    <DIV align=left><SPAN class=primera_letra>Y</SPAN><FONT
      face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\" size=2>a se ha
      comentado que la diferencia de potencial (tensión) es la causa de la
      circulación de los electrones en un circuito eléctrico. Por consiguiente,
      a mayor diferencia de potencial, mayor corriente de electrones.</FONT>
    </DIV>
    <P align=left><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
      size=2>También se ha visto que la resistencia eléctrica es la dificultad
      que ofrece un material al paso de la corriente eléctrica. Por lo tanto, si
      se mantiene constante la causa que hace circular a los electrones
      (diferencia de potencial), la corriente eléctrica dependerá de la
      dificultad que oponga el material a su paso, es decir, de su resistencia
      eléctrica.</FONT></P>
    <P align=left><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
      size=2>Experimentamente se comprueba que en un circuito eléctrico, la
      intensidad de la corriente que lo recorre es igual a la diferencia de
      potencial (d.d.p.) dividida por la resistencia del circuito.</FONT></P>
    <DIV align=left>
    <TABLE align=center bgColor=#ccccff border=1 cellPadding=10
      cellSpacing=0><TBODY>
      <TR>
        <TD><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
          size=2></FONT>
        <DIV align=center>
        <CENTER>
        <TABLE border=0 cellPadding=0 cellSpacing=0 height=55>
          <TBODY>
          <TR>
            <TD rowspan=3><STRONG><FONT face=\"Courier New\"
              size=4>Intensidad = </FONT></STRONG></TD>
            <TD rowspan=3><STRONG><FONT face=\"Courier New\" size=4>
              </FONT></STRONG></TD>
            <TD align=middle vAlign=center><STRONG><FONT
              face=\"Courier New\" size=4>Tensión
              (d.d.p.)</FONT></STRONG></TD></TR>
          <TR>
            <TD align=middle vAlign=center>

```



```

        <HR>
    </TD></TR>
<TR>
    <TD align=middle vAlign=center><STRONG><FONT
        face=\"Courier New\"
size=4>Resistencia</FONT></STRONG></TD></TR></TBODY></TABLE></CENTER></DIV></T
D></TR></TBODY></TABLE></DIV>
    <P align=left><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2>Del mismo modo:</FONT></P>
    <DIV align=left>
    <TABLE align=center bgColor=#ccccff border=1 cellPadding=10
        cellSpacing=0><TBODY>
    <TR>
    <TD>
        <P align=center><STRONG><FONT face=\"Courier New\" size=4>Tensión =
            Intensidad x Resistencia
    </FONT></STRONG></P></TD></TR></TBODY></TABLE></DIV>
    <P align=left><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2>y por último:</FONT></P>
    <DIV align=left>
    <TABLE align=center bgColor=#ccccff border=1 cellPadding=10
        cellSpacing=0><TBODY>
    <TR>
    <TD>
    <DIV align=center>
    <CENTER>
    <TABLE border=0 cellPadding=0 cellSpacing=0 height=55>
    <TBODY>
    <TR><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2><FONT
        face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2></FONT></FONT>
    <TD rowspan=3><STRONG><FONT face=\"Courier New\"
size=4>Resistencia = </FONT></STRONG></TD>
    <TD rowspan=3><STRONG><FONT face=\"Courier New\" size=4>
    </FONT></STRONG></TD>
    <TD align=middle vAlign=center><STRONG><FONT
        face=\"Courier new\"
size=4>Tensión</FONT></STRONG></TD></TR>
    <TR>
    <TD align=middle vAlign=center>
    <HR>
    </TD></TR>
    <TR>
    <TD align=middle vAlign=center><STRONG><FONT
        face=\"Courier New\"
size=4>Intensidad</FONT></STRONG></TD></TR></TBODY></TABLE></CENTER></DIV></TD
></TR></TBODY></TABLE>
    <P><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\" size=2>Esta
es la conocida como </FONT><SPAN class=superdestacado>Ley de
Ohm</SPAN><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2>, que relaciona los tres parámetros fundamentales de la
electricidad.</FONT></P></DIV><!-- #EndEditable -->
    <P align=right><A

```

```

href=\"http://members.es.tripod.de/electricoweb/cap0204.htm#Arribal\"></A>
</P></TD></TR></TBODY></TABLE>
<HR>

```

```

<DIV align=center>

```

```

<a href=\"../PsInteractividad/leyohm.html\">
Programa Interactivo de la Ley de Ohm</a>

```

```

<ul COMPACT><li><a href=http://www.yupi.com>Información Interesante</a><br>
<li><a href=http://www.cibercursos.com>Física en Línea</a><br>
</ul></font><hr>Preguntas...?, Comentarios...?<br> Enviar e-Mail a:

```

```

<address><a
href=mailto:salem.salem@grupopoma.com.sv>salem.salem@grupopoma.com.sv</address>
</font></BODY></HTML>', 'Ley de Ohm', 'salem.salem@grupopoma.com.sv',
'http://www.yupi.com', 'Información Interesante', 'http://www.cibercursos.com', 'Física en Línea', "", "",
",", "", "", "", "", "<bgsound src='../PsSonido/will_you_be_there[1].mid' loop='1'1'>

```

```

<STYLE type=text/css>A:active {
    COLOR: #ff0000; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
A:hover {
    COLOR: #009999; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
A:link {
    COLOR: #3333ff; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
A:visited {
    COLOR: #6666ff; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
.destacado {
    COLOR: #000066; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
.superdestacado {
    COLOR: #990000; FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif; FONT-SIZE: 10pt;
    FONT-WEIGHT: bold
}
.primera_letra {
    COLOR: #990000; FONT-FAMILY: \\"Times New Roman\\", Times, serif; FONT-SIZE: 18pt;
    FONT-WEIGHT: bold
}
.letra_destacada {
    COLOR: #ff3333; FONT-WEIGHT: bold
}
</STYLE>

```

```

<META content=\"none, default\" name=\"Microsoft Theme\">
<META content=\"none, default\" name=\"Microsoft Border\">
<META content=\"MSHTML 5.00.2314.1000\" name=GENERATOR></HEAD>

```

```

<BODY aLink=#cc0000 bgColor=#009eea link=#ff0000 text=#000000 vLink=#000000>
<P class=destacado><A name=Arriba></A></P>
<DIV align=center>
<CENTER>
<TABLE border=0 cellPadding=0 cellSpacing=0 width="100%">
  <TBODY>
    <TR bgColor=#000000 vAlign=center>
      <TD align=middle vAlign=center width="2%"><IMG height=30 hspace=10
        src="..\Pslmagen/atomo1.gif" width=30></TD>
      <TD align=middle vAlign=center
        width="94%"><!-- #BeginEditable "TEMA" --><FONT color=#ffffff
        face="Courier New, Courier, mono" size=5>Magnitudes eléctricas</FONT><!-- #EndEditable --
      </TD>
      <TD align=middle vAlign=center width="2%"><IMG height=30 hspace=10
        src="..\Pslmagen/atomo1.gif"
        width=30></TD></TR></TBODY></TABLE></CENTER></DIV>
<DIV align=center>
<CENTER>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="100%">
  <TBODY>
    <TR>
      <TD align=middle colSpan=3 vAlign=center><IMG height=8
        src="..\Pslmagen/transp.gif"
        width=8></TD></TR>
    <TR bgColor=#33ffff>
      <TD align=middle vAlign=baseline width=20> </TD>
      <TD align=left width="100%"><!-- #BeginEditable "APARTADO" --><FONT
        color=#000000 face="Courier New, Courier, mono" size=4>Ley de Ohm</FONT><!--
      #EndEditable --></TD>
      <TD align=middle vAlign=top width=20><IMG
        alt=Abajo border=0 height=20 hspace=20
        src="..\Pslmagen/Pabajo.gif"
        width=25></TD></TR></TBODY></TABLE></CENTER></DIV>
<TABLE align=center bgColor=#ffffff border=0 cellPadding=20 cellSpacing=0
width="100%">
  <TBODY>
    <TR bgColor=#ffffff>
      <TD><!-- #BeginEditable "TEXTO" -->
        <DIV align=left><SPAN class=primera_letra>Y</SPAN><FONT
        face="Verdana, Arial, Helvetica, Swiss, Geneva, Sans" size=2>a se ha
        comentado que la diferencia de potencial (tensión) es la causa de la
        circulación de los electrones en un circuito eléctrico. Por consiguiente,
        a mayor diferencia de potencial, mayor corriente de electrones.</FONT>
        </DIV>
        <P align=left><FONT face="Verdana, Arial, Helvetica, Swiss, Geneva, Sans"
        size=2>También se ha visto que la resistencia eléctrica es la dificultad
        que ofrece un material al paso de la corriente eléctrica. Por lo tanto, si
        se mantiene constante la causa que hace circular a los electrones
        (diferencia de potencial), la corriente eléctrica dependerá de la
        dificultad que oponga el material a su paso, es decir, de su resistencia
        eléctrica.</FONT></P>
        <P align=left><FONT face="Verdana, Arial, Helvetica, Swiss, Geneva, Sans"
        size=2>Experimentamente se comprueba que en un circuito eléctrico, la
        intensidad de la corriente que lo recorre es igual a la diferencia de
        potencial (d.d.p.) dividida por la resistencia del circuito.</FONT></P>
        <DIV align=left>

```



```

<TABLE align=center bgColor=#ccccff border=1 cellPadding=10
cellSpacing=0><TBODY>
<TR>
<TD><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2></FONT>
<DIV align=center>
<CENTER>
<TABLE border=0 cellPadding=0 cellSpacing=0 height=55>
<TBODY>
<TR>
<TD rowSpan=3><STRONG><FONT face=\"Courier New\"
size=4>Intensidad = </FONT></STRONG></TD>
<TD rowSpan=3><STRONG><FONT face=\"Courier New\" size=4>
</FONT></STRONG></TD>
<TD align=middle vAlign=center><STRONG><FONT
face=\"Courier New\" size=4>Tensión
(d.d.p.)</FONT></STRONG></TD></TR>
<TR>
<TD align=middle vAlign=center>
<HR>
</TD></TR>
<TR>
<TD align=middle vAlign=center><STRONG><FONT
face=\"Courier New\"
size=4>Resistencia</FONT></STRONG></TD></TR></TBODY></TABLE></CENTER></DIV></TD></TR></TBODY></TABLE></DIV>
<P align=left><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2>Del mismo modo:</FONT></P>
<DIV align=left>
<TABLE align=center bgColor=#ccccff border=1 cellPadding=10
cellSpacing=0><TBODY>
<TR>
<TD>
<P align=center><STRONG><FONT face=\"Courier New\" size=4>Tensión =
Intensidad x Resistencia
</FONT></STRONG></P></TD></TR></TBODY></TABLE></DIV>
<P align=left><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2>y por último:</FONT> </P>
<DIV align=left>
<TABLE align=center bgColor=#ccccff border=1 cellPadding=10
cellSpacing=0><TBODY>
<TR>
<TD>
<DIV align=center>
<CENTER>
<TABLE border=0 cellPadding=0 cellSpacing=0 height=55>
<TBODY>
<TR><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2><FONT
face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
size=2></FONT></FONT>
<TD rowSpan=3><STRONG><FONT face=\"Courier New\"
size=4>Resistencia = </FONT></STRONG></TD>
<TD rowSpan=3><STRONG><FONT face=\"Courier New\" size=4>
</FONT></STRONG></TD>

```

```

        <TD align=middle vAlign=center><STRONG><FONT
        face=\"Courier New\"
size=4>Tensión</FONT></STRONG></TD></TR>
    <TR>
        <TD align=middle vAlign=center>
            <HR>
        </TD></TR>
    <TR>
        <TD align=middle vAlign=center><STRONG><FONT
        face=\"Courier New\"
size=4>Intensidad</FONT></STRONG></TD></TR></TBODY></TABLE></CENTER></DIV></TD
></TR></TBODY></TABLE>
    <P><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\" size=2>Esta
    es la conocida como </FONT><SPAN class=superdestacado>Ley de
    Ohm</SPAN><FONT face=\"Verdana, Arial, Helvetica, Swiss, Geneva, Sans\"
    size=2>, que relaciona los tres parámetros fundamentales de la
    electricidad.</FONT></P></DIV><!-- #EndEditable -->
    <P align=right><A
    href=\"http://members.es.tripod.de/electricoweb/cap0204.htm#Arriba\"></A>
</P></TD></TR></TBODY></TABLE>
<HR>

```

```

<DIV align=center>

```

```

<a href=\"../PsInteractividad/leyohm.html\">
Programa Interactivo de la Ley de Ohm</a>

```

```

');
INSERT INTO PAGINAS VALUES( '0', '2', '2', '3', '2', '3', '#####', '2000-09-16', '2000-09-16',
'PAGINA', 'N', '3', '<HTML><HEAD><TITLE></TITLE></HEAD>
<CENTER><h1><font color=blue></font></h1></CENTER><hr>
<BODY><font size=2>
<HTML><HEAD><TITLE>Vectores</TITLE>
<META content=\"text/html; charset=iso-8859-1\" http-equiv=Content-Type>
<META content=\"MSHTML 5.00.2314.1000\" name=GENERATOR>
<META content=\"Definición de vectores\" name=Description>
<META content=\"vectores, cálculo vectorial\" name=KeyWords>
<script>
// end hide -->
</SCRIPT>
</HEAD>
<BODY bgColor=#ffffff>
<bgsound src=\"../PsSonido/thriller.mid\" loop=1\">
<HR SIZE=20 width=100%\">

```

```

<CENTER>
<H1><B><FONT face=Arial,Helvetica><FONT color=#008080>
Vectores</FONT></FONT></B></H1></CENTER>
<HR SIZE=20 width=100%\">

```

```

<P><FONT size=+1>En matemáticas, y por lo tanto en la física y la ingeniería, se
manejan tres tipos diferentes de cantidades. Éstas son <B><FONT
color=#008080>escalares</FONT></B><FONT color=#000000>, </FONT><B><FONT
color=#008080> vectores </FONT></B><FONT color=#000000>y</FONT><B><FONT

```

color=#008080> tensores.

<P> En este cuaderno estudiaremos los vectores y su álgebra.

<CENTER>

<p><TR></p> <p><TD bgcolor=#ffffcc>Un escalar es una cantidad que solo tiene una magnitud.

Un vector es una cantidad que tiene dos características: magnitud y dirección. </TD></TR></TABLE></CENTER>
 Ejemplos:

 Escalaes: masa, temperatura, área, longitud, dinero.
 Vectores: fuerza, desplazamiento, velocidad, aceleración, campo eléctrico.

 Para representar un vector, es costumbre utilizar una flecha.

 La longitud de la flecha es proporcional a la magnitud del vector y la orientación de la flecha indica la dirección del vector.
</p> <p></p> <p><CENTER></p> <table border="1" cellpadding="10" cellspacing="2" cols="1" width="50%"> <tr> <td> <p><TD></p> <p><CENTER></CENTER></TD></TR></TABLE></CENTER> </p> <p><CENTER> </CENTER></p> <p><CENTER></p> <table border="0" cellpadding="10" cellspacing="0" cols="1" width="80%"> <tr> <td> <p><TR></p> <p><TD>Notación:
 Para distinguir un vector de un escalar se denota a un vector con símbolos como: , <IMG alt="Vec</p> <p>tor B"</p> <p>height=20 src="..\Pslmagen/vector_B_azul.gif" width=15>, , etc.</TD></TR></TABLE></CENTER></p> <p><CENTER> </CENTER> </p> </td></tr></table></td></tr></table>	<p><TD></p> <p><CENTER></CENTER></TD></TR></TABLE></CENTER> </p> <p><CENTER> </CENTER></p> <p><CENTER></p> <table border="0" cellpadding="10" cellspacing="0" cols="1" width="80%"> <tr> <td> <p><TR></p> <p><TD>Notación:
 Para distinguir un vector de un escalar se denota a un vector con símbolos como: , <IMG alt="Vec</p> <p>tor B"</p> <p>height=20 src="..\Pslmagen/vector_B_azul.gif" width=15>, , etc.</TD></TR></TABLE></CENTER></p> <p><CENTER> </CENTER> </p> </td></tr></table>	<p><TR></p> <p><TD>Notación:
 Para distinguir un vector de un escalar se denota a un vector con símbolos como: , <IMG alt="Vec</p> <p>tor B"</p> <p>height=20 src="..\Pslmagen/vector_B_azul.gif" width=15>, , etc.</TD></TR></TABLE></CENTER></p> <p><CENTER> </CENTER> </p>
<p><TD></p> <p><CENTER></CENTER></TD></TR></TABLE></CENTER> </p> <p><CENTER> </CENTER></p> <p><CENTER></p> <table border="0" cellpadding="10" cellspacing="0" cols="1" width="80%"> <tr> <td> <p><TR></p> <p><TD>Notación:
 Para distinguir un vector de un escalar se denota a un vector con símbolos como: , <IMG alt="Vec</p> <p>tor B"</p> <p>height=20 src="..\Pslmagen/vector_B_azul.gif" width=15>, , etc.</TD></TR></TABLE></CENTER></p> <p><CENTER> </CENTER> </p> </td></tr></table>	<p><TR></p> <p><TD>Notación:
 Para distinguir un vector de un escalar se denota a un vector con símbolos como: , <IMG alt="Vec</p> <p>tor B"</p> <p>height=20 src="..\Pslmagen/vector_B_azul.gif" width=15>, , etc.</TD></TR></TABLE></CENTER></p> <p><CENTER> </CENTER> </p>	
<p><TR></p> <p><TD>Notación:
 Para distinguir un vector de un escalar se denota a un vector con símbolos como: , <IMG alt="Vec</p> <p>tor B"</p> <p>height=20 src="..\Pslmagen/vector_B_azul.gif" width=15>, , etc.</TD></TR></TABLE></CENTER></p> <p><CENTER> </CENTER> </p>		

```

<BR>
<HR width="100%">
<I><FONT color=#008080><FONT size=+3>Igualdad de vectores</FONT></FONT></I>
<BR> <BR>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
  <TBODY>
    <TR>
      <TD><FONT color=#800000><FONT size=+1>Definición:</FONT></FONT>
        <BR><FONT size=+1><FONT color=#800000> Dos vectores <IMG
          alt="Vector A" height=20 src="..\Pslmagen/vector_A_marron.gif"
          width=15> y <IMG alt="Vector B" height=20
          src="..\Pslmagen/vector_B_marron.gif" width=15> son
          iguales, <IMG alt="Vector A" height=20
          src="..\Pslmagen/vector_A_azul.gif" width=15> </FONT><FONT
          color=#0000ff>=</FONT><FONT color=#800000> <IMG alt="Vector B"
          height=20 src="..\Pslmagen/vector_B_azul.gif"
          width=15>,</FONT></FONT><FONT color=#800000> <FONT size=+1>si tienen la
          misma magnitud y la misma
          dirección.</FONT></FONT></TD></TR></TBODY></TABLE></CENTER> <BR><FONT
          color=#008080><FONT size=+1>Ejemplo:</FONT></FONT>
<UL>
  <UL>
    <CENTER>
      <TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
        <TBODY>
          <TR>
            <TD>
              <CENTER><IMG alt="Gráfica # 7 de vectores y el espacio tridimensional"
                height=185 src="..\Pslmagen/vect_graf_07.gif"
                width=300></CENTER></TD></TR>
            <TR>
              <TD>
                <CENTER><IMG alt="Vector AB" height=22
                  src="..\Pslmagen/vector_AB_marron.gif" width=28><I><FONT
                    color=#800000><FONT size=+1> = </FONT></FONT></I><IMG
                    alt="Vector CD" height=22
                    src="..\Pslmagen/vector_CD_marron.gif"
                    width=28></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL></UL>
            <H1>
              <HR width="100%">
              <I><FONT color=#008080>Definición de vectores en término de sus
              componentes</FONT></I></H1>
              <CENTER>
                <TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
                  <TBODY>
                    <TR>
                      <TD><FONT size=+1><FONT color=#800000>Algebraicamente se puede especificar
                      un vector como un par ordenado</FONT><FONT color=#3333ff>
                      <a,b></FONT><FONT color=#800000>. </FONT></FONT><FONT
                      color=#800000> </FONT> <BR><FONT color=#800000><FONT
                      size=+1>Los elementos del par ordenado se llaman componentes del
                      vector. </FONT></FONT></TD></TR></TBODY></TABLE></CENTER> <BR><FONT
                      color=#008080><FONT size=+1>Ejemplos:</FONT></FONT>
                    <UL>
                      <UL>

```

```

<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2>
<TBODY>
<TR>
<TD><IMG alt="Gráfica # 8 de vectores y el espacio tridimensional"
height=185 src="..\Pslmagen/vect_graf_08.gif"
width=300></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector A" height=17
src="..\Pslmagen/vector_Am_marron.gif" width=15><FONT
color=#800000><FONT size=+1><I> </I>= < </I>4, 2</I>
></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
<CENTER> </CENTER>
<CENTER> </CENTER>
<CENTER>
<TABLE border=1 cellPadding=10>
<TBODY>
<TR>
<TD><IMG alt="Gráfica # 9 de vectores y el espacio tridimensional"
height=185 src="..\Pslmagen/vect_graf_09.gif"
width=300></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector A" height=17
src="..\Pslmagen/vector_Am_marron.gif" width=15><FONT
color=#800000><FONT size=+1><I> </I>= < </I>2, 4</I>
></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
<CENTER> </CENTER></UL></UL>
<H1>
<HR width="100%">
<I><FONT color=#008080>Adición y sustracción</FONT></I></H1><FONT size=+1>La
suma de vectores se define mediante<FONT color=#008080> la ley del
paralelogramo</FONT>, que se ilustra enseguida.</FONT>
<UL>
<UL>
<TABLE border=1 cellPadding=10 cellSpacing=2>
<TBODY>
<TR>
<TD>
<CENTER><IMG
alt="Gráfica # 10 de vectores y el espacio tridimensional" height=300
src="..\Pslmagen/vect_graf_10.gif" width=240></CENTER></TD>
<TD>
<CENTER><IMG
alt="Gráfica # 11 de vectores y el espacio tridimensional" height=225
src="..\Pslmagen/vect_graf_11.gif"
width=300></CENTER></TD></TR>
<TR>
<TD>
<CENTER><FONT color=#800000><FONT size=+1>Vectores
originales</FONT></FONT></CENTER></TD>
<TD>
<CENTER><FONT color=#800000><FONT size=+1>Vectores
desplazados</FONT></FONT></CENTER></TD></TR></TBODY></TABLE> </UL></UL>
<UL>

```



```

<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
  <TBODY>
    <TR>
      <TD>
        <CENTER><IMG
          alt="Gráfica # 12 de vectores y el espacio tridimensional" height=300
          src="..\Pslmagen/vect_graf_12.gif"
width=200></CENTER></TD></TR>
    <TR>
      <TD>
        <CENTER><FONT color=#800000><FONT size=+1>Vectores y su
suma</FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL> </UL><F
ONT
size=+1> Aunque hemos ilustrado a los vectores en un plano,
ahora definiremos a los vectores y sus operaciones en el espacio
tridimensional.</FONT> <BR> <BR>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
  <TBODY>
    <TR>
      <TD><FONT color=#800000><FONT size=+1>En general, un vector <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_marron.gif"
width=15> en el espacio tridimensional es cualquier triada de números
reales,</FONT> </FONT>
<CENTER><IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_azul.gif" width=15><FONT
size=+1><FONT color=#800000> </FONT><FONT color=#0000ff>=
<a<SUB>1</SUB>, a<SUB>2</SUB>, a<SUB>3</SUB></FONT></FONT><FONT
color=#0000ff> </FONT></CENTER><FONT size=+1><FONT color=#800000>en
donde los números </FONT><FONT color=#0000ff>a<SUB>1</SUB>, a<SUB>2</SUB>,
a<SUB>3</SUB></FONT><FONT color=#800000> se llaman </FONT><FONT
color=#0000ff>componentes</FONT><FONT color=#800000> del vector <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_azul.gif"
width=15></FONT></FONT><FONT
color=#800000>. </FONT> </TD></TR></TBODY></TABLE></CENTER>
<BR><FONT size=+1> <FONT color=#008080>
Ejemplo:</FONT></FONT> <BR>
<UL>
<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
  <TBODY>
    <TR>
      <TD>
        <CENTER><IMG
          alt="Gráfica # 13 de vectores y el espacio tridimensional" height=300
          src="..\Pslmagen/vect_graf_13.gif"
width=246></CENTER></TD></TR>
    <TR>
      <TD>
        <CENTER><IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_marron.gif" width=15><FONT
face="Times New Roman,Times"><FONT color=#800000><FONT size=+1> =

```

{<l>4, 2,

3</l>}</CENTER></TD></TR></TBODY></TABLE></CENTER> <FONT

color=#000000> En términos de componentes, la suma de vectores se define como sigue:

<CENTER>

<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width=1"80%">

<TBODY>

<TR>

<TD> Sean = <x₁, y₁, z₁> y = <x₂, y₂, z₂>, la suma de y se define como: <CENTER> + = <x₁+ x₂, y₁+ y₂, z₁+ z₂> </CENTER></TD></TR></TBODY></TABLE></CENTER>

Ejemplo:

<CENTER>

<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width=1"50%">

<TBODY>

<TR>

<TD>

<CENTER><IMG

alt="Gráfica # 14 de vectores y el espacio tridimensional" height=270

src="..\Pslmagen/vect_graf_14.gif"

width=300></CENTER></TD></TR>

<TR>

<TD>

<CENTER><IMG alt="Vector A" height=20

src="..\Pslmagen/vector_A_marron.gif" width=15><FONT

face="Times New Roman,Times"><l>

</l>= { <l>3, 2, 3</l>} <IMG alt="Vector B" height=20

src="..\Pslmagen/vector_B_marron.gif" width=15> = {<l>2,

2, 0</l>}</CENTER></TD></TR>

<TR>

<TD>

<CENTER><IMG alt="Vector C" height=20

src="..\Pslmagen/vector_C_marron.gif" width=15><FONT

face="Times New Roman,Times">


```
= <IMG alt=\"Vector A\" height=20
src=\"../Pslimagen/vector_A_marron.gif\" width=15></I>
+ <IMG alt=\"Vector B\" height=20
src=\"../Pslimagen/vector_B_marron.gif\" width=15></I> =
{</I>5, 4,
```

```
3</I>}</FONT></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL>
></UL> <FONT
```

```
color=#000000> <FONT size=+1>Y la igualdad entre vectores se define entonces de
la siguiente manera:</FONT> </FONT> <BR>
```

```
<CENTER>
```

```
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width=\"70%\">
```

```
<TBODY>
```

```
<TR>
```

```
<TD><FONT size=+1> <FONT color=#3333ff> </FONT><FONT
```

```
color=#800000>Sean <IMG alt=\"Vector A\" height=20
```

```
src=\"../Pslimagen/vector_A_marron.gif\" width=15> =
```

```
<a<SUB>1</SUB>, a<SUB>2</SUB>, a<SUB>3</SUB>> y <IMG
```

```
alt=\"Vector B\" height=20 src=\"../Pslimagen/vector_B_marron.gif\"
```

```
width=15> = <b<SUB>1</SUB>, b<SUB>2</SUB>, b<SUB>3</SUB>>,
```

```
entonces <IMG alt=\"Vector A\" height=20
```

```
src=\"../Pslimagen/vector_A_azul.gif\" width=15> = <IMG
```

```
alt=\"Vector A\" height=20 src=\"../Pslimagen/vector_B_azul.gif\"
```

```
width=15> si y sólo si</FONT></FONT><FONT
```

```
color=#800000> </FONT>
```

```
<CENTER><FONT color=#800000> </FONT><FONT color=#0000ff><FONT
```

```
size=+1>a<SUB>1</SUB>= b<SUB>1</SUB>, a<SUB>2</SUB>= b<SUB>2</SUB>,
```

```
a<SUB>3</SUB>= b<SUB>3</SUB></FONT></FONT><FONT
```

```
color=#800000> </FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
```

```
<BR>
```

```
<H1>
```

```
<HR width=\"100%\">
```

```
<I><FONT color=#008080>La magnitud de un vector en términos de sus
componentes.</FONT></I></H1> <BR> <FONT size=+1>Por el
```

```
teorema de <FONT color=#008080>Pitágoras</FONT>,</FONT> <BR><FONT
size=+1> </FONT>
```

```
<CENTER>
```

```
<TABLE bgColor=#993366 cellPadding=10 cellSpacing=0 cols=1 width=\"80%\">
```

```
<TBODY>
```

```
<TR>
```

```
<TD><FONT color=#ffffff><FONT size=+2> </FONT><FONT
```

```
size=+1>Sea <IMG alt=\"Vector A\" height=20
```

```
src=\"../Pslimagen/vector_A_blanco.gif\" width=15> =
```

```
<a<SUB>1</SUB>,
```

```
a<SUB>2</SUB>,a<SUB>3</SUB>>,entonces </FONT></FONT>
```

```
<CENTER><FONT color=#ffffff><B>|<IMG alt=\"Vector A\" height=20
```

```
src=\"../Pslimagen/vector_A_blanco.gif\" width=15></B><FONT
```

```
size=+1>| = ( a<SUB>1</SUB><SUP>2</SUP> + a<SUB>2</SUB><SUP>2</SUP> +
```

```
a<SUB>3</SUB><SUP>2</SUP>
```

```
)<SUP>1/2</SUP></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
```

```
<BR> <FONT color=#000000><FONT size=+1>Ahora definiremos
```

```
otra operación, la multiplicación de un vector por un
```

```
escalar. </FONT></FONT> <BR><FONT color=#000000> </FONT>
```

```
<CENTER>
```

```
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width=\"80%\">
```

```
<TBODY>
```

```

<TR>
<TD> <FONT size=+1><FONT color=#3333ff> </FONT><FONT
color=#800000>Sea <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_marron.gif" width=15> =
<a<SUB>1</SUB>, a<SUB>2</SUB>, a<SUB>3</SUB>>> y k un escalar,
entonces definimos la multiplicación por un escalar como
sigue:</FONT></FONT> <BR><FONT color=#800000><FONT
size=+1> </FONT></FONT>
<CENTER><FONT color=#800000><FONT size=+1>k <IMG alt="Vector A"
height=20 src="..\Pslmagen/vector_A_marron.gif" width=15> =
<k a<SUB>1</SUB>, k a<SUB>2</SUB>, k
a<SUB>3</SUB>>> </FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
<BR> <BR><FONT size=+1> <FONT
color=#008080> Ejemplo:</FONT></FONT>
<UL>
<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
<TBODY>
<TR>
<TD>
<CENTER><IMG
alt="Gráfica # 15 de vectores y el espacio tridimensional" height=241
src="..\Pslmagen/vect_graf_15.gif"
width=288></CENTER></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_marron.gif" width=15><FONT
face="Times New Roman, Times" ><FONT color=#800000><FONT size=+1><|>
</|>= {<|>4, 3, 2</|>} <|>k = 1 /
4</|></FONT></FONT></FONT></CENTER></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector C" height=20
src="..\Pslmagen/vector_C_marron.gif" width=15><|><FONT
face="Times New Roman, Times" ><FONT color=#800000><FONT size=+1> =
k <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_marron.gif" width=15> = {1, 3 /
4, 1 /
2}</FONT></FONT></FONT></|></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL>
</UL>
<BR> <FONT size=+1> Obsérvese que <FONT color=#008080>k<IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_verde.gif"
width=15></FONT> es un vector con la misma dirección que <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_verde.gif"
width=15> si <FONT color=#008080>k > 0</FONT> y con dirección contraria si
<FONT color=#008080>k < 0</FONT>. </FONT> <BR><FONT size=+1>
La magnitud de <FONT color=#008080>k<IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_verde.gif" width=15></FONT> es <FONT
color=#008080>|k| <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_verde.gif" width=15>|</FONT> .</FONT>
<BR> <BR>
<CENTER>

```

```

<TABLE bgColor=#ffffcc border=0 cellPadding=10 cols=1 width="80%">
<TBODY>
<TR>
<TD><FONT size=+1><FONT color=#800000>En virtud de lo anterior,
</FONT><FONT color=#0000ff>el negativo de</FONT><FONT
color=#800000> <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_azul.gif"
width=15></FONT></FONT><FONT color=#800000> <FONT
size=+1>es</FONT></FONT>
<CENTER><FONT color=#0000ff><FONT size=+1>- <IMG alt="Vector A"
height=20 src="..\Pslmagen/vector_A_azul.gif" width=15> =
(-1) </FONT></FONT><IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_azul.gif"
width=15></CENTER></TD></TR></TBODY></TABLE></CENTER>
<H1>
<HR width="100%">
<I><FONT color=#008080>La sustracción o resta de vectores.</FONT></I></H1>

```

```

<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
<TBODY>
<TR>
<TD><FONT color=#3333ff><FONT size=+1>Se define la resta de vectores
como: </FONT> </FONT>
<CENTER><IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_azul.gif" width=15><FONT
color=#3333ff><FONT size=+1> - <IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_azul.gif" width=15> = <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_azul.gif"
width=15> + (- <IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_azul.gif"
width=15>) </FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>

```

```

<H1>
<HR width="100%">
<I><FONT color=#008080>Propiedades:</FONT></I></H1><FONT
size=+1> Con la definición de vector en términos de
componentes y con las</FONT> <FONT size=+1>propiedades de los números reales,
puede demostrarse el siguiente teorema.</FONT> <BR>

```

```

<CENTER>
<TABLE bgColor=#993366 cellPadding=10 cellSpacing=0 cols=1 width="80%">
<TBODY>
<TR>
<TD><FONT color=#ffffff><FONT size=+2>Teorema</FONT></FONT>
<BR><FONT color=#ffffff><FONT size=+1>Sean <IMG alt="Vector A"
height=20 src="..\Pslmagen/vector_A_blanco.gif" width=15>
y <IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_blanco.gif" width=15> dos vectores y
k<SUB>1</SUB> y k<SUB>2</SUB> dos escalares.
Entonces:</FONT> </FONT> <BR> <BR> <FONT
color=#ffffff><FONT size=+1>i) <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15> + <IMG
alt="Vector B" height=20 src="..\Pslmagen/vector_B_blanco.gif"
width=15> = <IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_blanco.gif" width=15> + <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_blanco.gif"

```

```

width=15>
Ley conmutativa</FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>ii) <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15> + (<IMG
alt="Vector B" height=20 src="\"../Pslmagen/vector_B_blanco.gif"
width=15> + <IMG alt="Vector C" height=20
src="\"../Pslmagen/vector_C_blanco.gif" width=15>) = (<IMG
alt="Vector A" height=20 src="\"../Pslmagen/vector_A_blanco.gif"
width=15> + <IMG alt="Vector B" height=20
src="\"../Pslmagen/vector_B_blanco.gif" width=15>) + <IMG
alt="Vector C" height=20 src="\"../Pslmagen/vector_C_blanco.gif"
width=15> Ley asociativa</FONT> </FONT>
<BR><FONT color=#ffffff><FONT size=+1>iii) <IMG alt="Vector A"
height=20 src="\"../Pslmagen/vector_A_blanco.gif" width=15>
+ <IMG alt="Vector O" height=20
src="\"../Pslmagen/vector_O_blanco.gif" width=15> = <IMG
alt="Vector A" height=20 src="\"../Pslmagen/vector_A_blanco.gif"
width=15>
Identidad aditiva</FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>iv) <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15> + (<IMG
alt="Vector A" height=20 src="\"../Pslmagen/vector_A_blanco.gif"
width=15>) = <IMG alt="Vector O" height=20
src="\"../Pslmagen/vector_O_blanco.gif"
width=15>
Inverso aditivo</FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>v) k<SUB>1</SUB>(<IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15> + <IMG
alt="Vector B" height=20 src="\"../Pslmagen/vector_B_blanco.gif"
width=15>) = k<SUB>1<IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15></SUB> +
k<SUB>1<IMG alt="Vector B" height=20
src="\"../Pslmagen/vector_B_blanco.gif"
width=15></SUB></FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>vi) (k<SUB>1 </SUB>+ k<SUB>2</SUB>) <IMG alt="Vector A"
height=20 src="\"../Pslmagen/vector_A_blanco.gif" width=15> =
k<SUB>1<IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15></SUB> +
k<SUB>2<IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif"
width=15></SUB></FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>vii) k<SUB>1</SUB>( k<SUB>2<IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15></SUB>) =
(k<SUB>1</SUB>k<SUB>2</SUB>) <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif"
width=15></FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>viii) 1 <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15> = <IMG
alt="Vector A" height=20 src="\"../Pslmagen/vector_A_blanco.gif"
width=15></FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>ix) 0 <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15> = <IMG
alt="Vector O" height=20 src="\"../Pslmagen/vector_O_blanco.gif"
width=15></FONT> </FONT></TD></TR></TBODY></TABLE></CENTER>
<H1>
<HR width="100%">

```

```

<I><FONT color=#008080>Los vectores unitarios <IMG align=ABSCENTER
alt=\"Vector i unitario\" height=30
src=\"../Pslmagen/vector_i_verde.gif\" width=18>, <IMG align=ABSC
alt=\"Vector j unitario\" height=30
src=\"../Pslmagen/vector_j_verde.gif\" width=18 ENTER> y <IMG
align=ABSCENTER alt=\"Vector k unitario\" height=30
src=\"../Pslmagen/vector_k_verde.gif\" width=18>.</FONT></I></H1><FONT
size=+1>Cualquier vector <IMG alt=\"Vector A\" height=20
src=\"../Pslmagen/vector_A_verde.gif\" width=15><B> <IMG
alt=Distinto height=15 src=\"../Pslmagen/distint_verde.gif\"
width=13> <IMG alt=\"Vector O\" height=20
src=\"../Pslmagen/vector_O_verde.gif\" width=15></B> dá origen a un
vector<B> <IMG alt=\"Vector A\" height=17
src=\"../Pslmagen/vector_Amu_verde.gif\"></B> con la misma dirección
pero de magnitud 1, de la siguiente manera:</FONT> <BR> <BR>
<CENTER>
<TABLE border=0>
  <TBODY>
    <TR>
      <TD></TD>
      <TD></TD>
      <TD>
        <CENTER><FONT color=#008080><FONT size=+1>1</FONT></FONT></CENTER></TD>
      <TD></TD></TR>
    <TR>
      <TD><IMG alt=\"Vector unitario a\" height=17
        src=\"../Pslmagen/vector_Amu_verde.gif\" width=15></TD>
      <TD><FONT color=#008080><FONT size=+1>=</FONT></FONT></TD>
      <TD>
        <HR noShade width=\"100%\">
      </TD>
      <TD><IMG align=ABSCENTER alt=\"Vector A\" height=20
        src=\"../Pslmagen/vector_A_verde.gif\" width=15></TD></TR>
    <TR>
      <TD></TD>
      <TD></TD>
      <TD><FONT color=#008080>|<IMG align=ABSCENTER alt=\"Vector A\" height=20
        src=\"../Pslmagen/vector_A_verde.gif\" width=15>|</FONT></TD>
      <TD></TD></TR></TBODY></TABLE></CENTER>
<P><FONT color=#008080><FONT size=+1>Ejemplo:</FONT></FONT>
<UL>
  <UL>
    <CENTER>
      <TABLE border=1 cellPadding=10 cellSpacing=2>
        <TBODY>
          <TR>
            <TD>
              <CENTER><IMG alt=\"Vector A\" height=20
                src=\"../Pslmagen/vector_A_marron.gif\" width=15><FONT
                color=#800000><FONT size=+1> = {3, 1, 4}
                | <IMG alt=\"Vector A\" height=20
                src=\"../Pslmagen/vector_A_marron.gif\" width=15> | =
                (26)<SUP>1/2</SUP></FONT></FONT></CENTER></TD></TR>
          <TR>
            <TD>
              <CENTER><IMG alt=\"Vector unitario a\" height=17

```



```
src=\"../Pslimagen/vector_Amu_marron.gif\" width=15><FONT
color=#800000><FONT size=+1> = {3/(26)<SUP>1/2</SUP>,
1/(26)<SUP>1/2</SUP>, 4/(26)<SUP>1/2</SUP>}
| <IMG alt=\"Vector unitario a\" height=17
src=\"../Pslimagen/vector_Amu_marron.gif\" width=15> | =
```

```
1</FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL></UL><FONT
size=+1> Por las definiciones dadas anteriormente, cualquier
vector</FONT> <BR> <BR>
```

```
<CENTER>
```

```
<TABLE border=0>
```

```
<TBODY>
```

```
<TR>
```

```
<TD><IMG alt=\"Vector A\" height=20
```

```
src=\"../Pslimagen/vector_A_verde.gif\" width=15><FONT
```

```
color=#008080> <FONT size=+1>= <a<SUB>1</SUB>, a<SUB>2</SUB>,
a<SUB>3</SUB></FONT></FONT></TD></TR></TBODY></TABLE></CENTER>
```

```
<FONT size=+1>se puede escribir en la forma</FONT> <BR> <BR>
```

```
<CENTER>
```

```
<TABLE bgColor=#ffffcc border=0>
```

```
<TBODY>
```

```
<TR>
```

```
<TD><IMG alt=\"Vector A\" height=20
```

```
src=\"../Pslimagen/vector_A_marron.gif\" width=15><FONT
```

```
color=#800000> <FONT size=+1>= a<SUB>1</SUB><1, 0, 0> +
```

```
a<SUB>2</SUB><0, 1, 0> + a<SUB>3</SUB><0, 0,
```

```
1></FONT></FONT></TD></TR></TBODY></TABLE></CENTER>
```

```
<CENTER>
```

```
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width=\"80%\">
```

```
<TBODY>
```

```
<TR>
```

```
<TD><FONT color=#800000><FONT
```

```
size=+1>Definición:</FONT> </FONT> <BR><FONT
```

```
color=#800000><FONT size=+1>Definimos los vectores
```

```
unitarios</FONT></FONT><FONT color=#3333ff> </FONT>
```

```
<CENTER><IMG alt=\"Vector i unitario\" height=20
```

```
src=\"../Pslimagen/vector_i_azul.gif\" width=12><FONT
```

```
color=#3333ff><FONT size=+1> = <1,0,0></FONT> </FONT></CENTER>
```

```
<CENTER><IMG alt=\"Vector j unitario\" height=20
```

```
src=\"../Pslimagen/vector_j_azul.gif\" width=12><FONT
```

```
color=#3333ff><FONT size=+1> = <0,1,0></FONT> </FONT></CENTER>
```

```
<CENTER><IMG alt=\"Vector k unitario\" height=20
```

```
src=\"../Pslimagen/vector_k_azul.gif\" width=12><FONT
```

```
color=#3333ff><FONT size=+1> =
```

```
<0,0,1></FONT> </FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
```

```
<BR><FONT size=+1> Entonces, por lo anterior, cualquier
```

```
vector <IMG alt=\"Vector A\" height=20
```

```
src=\"../Pslimagen/vector_A_verde.gif\" width=15> se puede expresar en
```

```
la forma</FONT> <BR> <BR>
```

```
<CENTER>
```

```
<TABLE border=0>
```

```
<TBODY>
```

```
<TR>
```

```
<TD><IMG alt=\"Vector A\" height=20
```

```
src=\"../Pslimagen/vector_A_verde.gif\" width=15><FONT
```

```
color=#008080><FONT size=+1> = a<SUB>1<IMG align=ABSCENTER
```

```

alt="Vector i unitario" height=20
src="..\Pslmagen/vector_i_verde.gif" width=12></SUB> +
a<SUB>2<IMG align=ABSCENTER alt="Vector j unitario" height=20
src="..\Pslmagen/vector_i_verde.gif" width=12></SUB> +
a<SUB>3</SUB></FONT></FONT><IMG alt="Vector k unitario" height=20
src="..\Pslmagen/vector_k_verde.gif"
width=12></TD></TR></TBODY></TABLE></CENTER><FONT size=+1> </FONT>
<BR> <FONT size=+1> A los vectores <FONT
color=#008080>a<SUB>1<IMG alt="Vector i unitario" height=20
src="..\Pslmagen/vector_i_verde.gif" width=12></SUB></FONT><FONT
color=#000000>,</FONT><FONT color=#008080><SUB> </SUB>a<SUB>2<IMG
alt="Vector j unitario" height=20
src="..\Pslmagen/vector_i_verde.gif" width=12></SUB></FONT>,<FONT
color=#008080>a<SUB>3<IMG alt="Vector k unitario" height=20
src="..\Pslmagen/vector_k_verde.gif" width=12></SUB></FONT> se les
llama <FONT color=#008080>vectores componentes</FONT>.</FONT> <BR>
<UL>
<UL>
<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2>
<TBODY>
<TR>
<TD>
<CENTER><FONT color=#008080><FONT size=+1>Vectores
unitarios <IMG alt="Vector i unitario" height=20
src="..\Pslmagen/vector_i_verde.gif"
width=12>,<IMG alt="Vector j unitario" height=20
src="..\Pslmagen/vector_i_verde.gif"
width=12>,</FONT></FONT><IMG alt="Vector k unitario" height=20
width=12 C="..\vector_k_verde.gif" SR></CENTER></TD></TR>
<TR>
<TD><IMG alt="Gráfica # 16 de vectores y el espacio tridimensional"
height=288 src="..\Pslmagen/vect_graf_16.gif"
width=265></TD></TR></TBODY></TABLE></CENTER> </UL></UL></UL> <BR>
<HR width="100%">
<a href="..\PsInteractividad/resultesp.html">
Composición de Fuerzas (Programa Interactivo)</a>

```

```

<ul COMPACT><li><a href=http://www.yupi.com>Información Interesante</a><br>
<li><a href=http://www.cibercursos.com>Física en Linea</a><br>
</ul></font><hr>Preguntas...?, Comentarios...?<br> Enviar e-Mail a:
<address><a
href=mailto:salem.salem@grupopoma.com.sv>salem.salem@grupopoma.com.sv</address>
</font></BODY></HTML>','salem.salem@grupopoma.com.sv','http://www.yupi.com',
'Información Interesante','http://www.cibercursos.com','Física en Linea',' ',' ',' ',' ',' ',' ',' '
'<HTML><HEAD><TITLE>Vectores</TITLE>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<META content="MSHTML 5.00.2314.1000" name=GENERATOR>
<META content="Definición de vectores" name=Description>
<META content="vectores, cálculo vectorial" name=KeyWords>
<script>
// end hide -->

```



```
</SCRIPT>
</HEAD>
<BODY bgColor=#ffffff>
<bgsound src=\"../PsSonido/thriller.mid\" loop=\"1\">
<HR SIZE=20 width=\"100%\">
```

```
<CENTER>
<H1><B><FONT face=Arial,Helvetica><FONT color=#008080>
Vectores</FONT></FONT></B></H1></CENTER>
<HR SIZE=20 width=\"100%\">
```

```
<P><FONT size=+1>En matemáticas, y por lo tanto en la física y la ingeniería, se
manejan tres tipos diferentes de cantidades. Éstas son <B><FONT
color=#008080>escalares</FONT></B><FONT color=#000000>,</FONT><B><FONT
color=#008080> vectores </FONT></B><FONT color=#000000>y</FONT><B><FONT
color=#008080> tensores</FONT></B></FONT>
```

```
<P><FONT size=+1> En este cuaderno estudiaremos los vectores y
su álgebra.</FONT> <BR> <BR>
```

```
<CENTER>
<TABLE border=0 cellPadding=10 cellSpacing=0 cols=1 width=\"80%\">
<TBODY>
```

```
<TR>
<TD bgColor=#ffffcc><FONT color=#800000><FONT size=+1>Un escalar es una
cantidad que solo tiene una magnitud.</FONT> </FONT> <BR><FONT
color=#800000> </FONT> <BR><FONT size=+1><FONT color=#800000>Un
</FONT><FONT color=#0000ff>vector</FONT><FONT color=#800000> es una
cantidad que tiene dos características:</FONT></FONT><FONT color=#800000>
</FONT><FONT size=+1><FONT color=#0000ff>magnitud</FONT><FONT
color=#800000> y</FONT></FONT><FONT color=#800000> </FONT><FONT
size=+1><FONT color=#0000ff>dirección</FONT><FONT
color=#800000>.</FONT></FONT><FONT
```

```
color=#800000> </FONT></TD></TR></TBODY></TABLE></CENTER><FONT
size=+1> </FONT> <BR><FONT size=+1> <FONT
color=#008080>Ejemplos:</FONT></FONT> <BR> <BR><FONT
size=+1> <FONT color=#008080>Escalares</FONT>: masa,
temperatura, área, longitud, dinero.</FONT> <BR><FONT size=+1>
<FONT color=#008080>Vectores</FONT>: fuerza, desplazamiento, velocidad,
aceleración, campo eléctrico.</FONT> <BR> <BR><FONT
size=+1> Para representar un vector, es costumbre utilizar una
flecha.</FONT> <BR> <BR><FONT size=+1> La longitud de la
flecha es proporcional a la magnitud del vector y la orientación de la flecha
indica la dirección del vector.</FONT> <BR>
```

```
<UL>
```

```
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width=\"50%\">
<TBODY>
<TR>
<TD>
<CENTER><IMG alt=\"Gráfica # 6 de vectores y el espacio tridimensional\"
height=185 src=\"../PsImagen/vect_graf_06.gif\"
width=300></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL>
```

```
<CENTER> </CENTER>
```

```
<CENTER>
```

```
<TABLE bgColor=#ffffcc border=0 cellPadding=10 cellSpacing=0 cols=1
width=\"80%\"><TBODY>
<TR>
```

```

<TD><FONT size=+1><FONT color=#800000>Notación:</FONT> </FONT>
<BR><FONT size=+1> <FONT color=#800000>Para distinguir un
vector de un escalar se denota a un vector con símbolos
como: </FONT></FONT><FONT color=#800000> <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_azul.gif"
width=15><FONT size=+1>, <IMG alt="Vec
tor B"
height=20 src="..\Pslmagen/vector_B_azul.gif"
width=15>, <IMG alt="Vector AB" height=22
src="..\Pslmagen/vector_AB_azul.gif" width=28>,
etc.</FONT></FONT></TD></TR></TBODY></TABLE></CENTER>
<CENTER> </CENTER><FONT color=#800000><FONT size=+2> </FONT></FONT>
<BR>
<HR width="100%">
<I><FONT color=#008080><FONT size=+3>Igualdad de vectores</FONT></FONT></I>
<BR> <BR>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
<TBODY>
<TR>
<TD><FONT color=#800000><FONT size=+1>Definición:</FONT></FONT>
<BR><FONT size=+1><FONT color=#800000> Dos vectores <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_marron.gif"
width=15> y <IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_marron.gif" width=15> son
iguales, <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_azul.gif" width=15> </FONT><FONT
color=#0000ff>=</FONT><FONT color=#800000> <IMG alt="Vector B"
height=20 src="..\Pslmagen/vector_B_azul.gif"
width=15>,</FONT></FONT><FONT color=#800000> <FONT size=+1>si tienen la
misma magnitud y la misma
dirección.</FONT></FONT></TD></TR></TBODY></TABLE></CENTER> <BR><FONT
color=#008080><FONT size=+1>Ejemplo:</FONT></FONT>
<UL>
<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
<TBODY>
<TR>
<TD>
<CENTER><IMG alt="Gráfica # 7 de vectores y el espacio tridimensional"
height=185 src="..\Pslmagen/vect_graf_07.gif"
width=300></CENTER></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector AB" height=22
src="..\Pslmagen/vector_AB_marron.gif" width=28><I><FONT
color=#800000><FONT size=+1> = </FONT></FONT></I><IMG
alt="Vector CD" height=22
src="..\Pslmagen/vector_CD_marron.gif"
width=28></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL></UL>
<H1>
<HR width="100%">

```

```

<I><FONT color=#008080>Definición de vectores en término de sus
componentes</FONT></I></H1>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
  <TBODY>
    <TR>
      <TD><FONT size=+1><FONT color=#800000>Algebraicamente se puede especificar
        un vector como un par ordenado</FONT><FONT color=#3333ff>
        <a,b></FONT><FONT color=#800000>. </FONT></FONT><FONT
        color=#800000> </FONT> <BR><FONT color=#800000><FONT
        size=+1>Los elementos del par ordenado se llaman componentes del
        vector. </FONT></FONT></TD></TR></TBODY></TABLE></CENTER> <BR><FONT
        color=#008080><FONT size=+1>Ejemplos:</FONT></FONT>
<UL>
<UL>
  <CENTER>
    <TABLE border=1 cellPadding=10 cellSpacing=2>
      <TBODY>
        <TR>
          <TD><IMG alt="Gráfica # 8 de vectores y el espacio tridimensional"
            height=185 src="\"../Pslmagen/vect_graf_08.gif"
            width=300></TD></TR>
          <TR>
            <TD>
              <CENTER><IMG alt="Vector A" height=17
                src="\"../Pslmagen/vector_Am_marron.gif" width=15><FONT
                color=#800000><FONT size=+1><I> </I>= < <I>4, 2</I>
                ></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
            <CENTER> </CENTER>
            <CENTER> </CENTER>
            <CENTER>
              <TABLE border=1 cellPadding=10>
                <TBODY>
                  <TR>
                    <TD><IMG alt="Gráfica # 9 de vectores y el espacio tridimensional"
                      height=185 src="\"../Pslmagen/vect_graf_09.gif"
                      width=300></TD></TR>
                    <TR>
                      <TD>
                        <CENTER><IMG alt="Vector A" height=17
                          src="\"../Pslmagen/vector_Am_marron.gif" width=15><FONT
                          color=#800000><FONT size=+1><I> </I>= < <I>2, 4</I>
                          ></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
                        <CENTER> </CENTER></UL></UL>
<H1>
<HR width="100%">
<I><FONT color=#008080>Adición y sustracción</FONT></I></H1><FONT size=+1>La
suma de vectores se define mediante<FONT color=#008080> la ley del
paralelogramo</FONT>, que se ilustra enseguida.</FONT>
<UL>
<UL>
  <TABLE border=1 cellPadding=10 cellSpacing=2>
    <TBODY>
      <TR>
        <TD>
          <CENTER><IMG

```

```

alt="Gráfica # 10 de vectores y el espacio tridimensional" height=300
src="..\Pslmagen/vect_graf_10.gif" width=240></CENTER></TD>
<CENTER><IMG
alt="Gráfica # 11 de vectores y el espacio tridimensional" height=225
src="..\Pslmagen/vect_graf_11.gif"
width=300></CENTER></TD></TR>
<TR>
<TD>
<CENTER><FONT color=#800000><FONT size=+1>Vectores
originales</FONT></FONT></CENTER></TD>
<TD>
<CENTER><FONT color=#800000><FONT size=+1>Vectores
desplazados</FONT></FONT></CENTER></TD></TR></TBODY></TABLE> </UL></UL>
<UL>
<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
<TBODY>
<TR>
<TD>
<CENTER><IMG
alt="Gráfica # 12 de vectores y el espacio tridimensional" height=300
src="..\Pslmagen/vect_graf_12.gif"
width=200></CENTER></TD></TR>
<TR>
<TD>
<CENTER><FONT color=#800000><FONT size=+1>Vectores y su
suma</FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL> </UL><F
ONT
size=+1> Aunque hemos ilustrado a los vectores en un plano,
ahora definiremos a los vectores y sus operaciones en el espacio
tridimensional.</FONT> <BR> <BR>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
<TBODY>
<TR>
<TD><FONT color=#800000><FONT size=+1>En general, un vector <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_marron.gif"
width=15> en el espacio tridimensional es cualquier tríada de números
reales,</FONT> </FONT>
<CENTER><IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_azul.gif" width=15><FONT
size=+1><FONT color=#800000> </FONT><FONT color=#0000ff>=
<a<SUB>1</SUB>, a<SUB>2</SUB>, a<SUB>3</SUB>></FONT></FONT><FONT
color=#0000ff> </FONT></CENTER><FONT size=+1><FONT color=#800000>en
donde los números </FONT><FONT color=#0000ff>a<SUB>1</SUB>, a<SUB>2</SUB>,
a<SUB>3</SUB></FONT><FONT color=#800000> se llaman </FONT><FONT
color=#0000ff>componentes</FONT><FONT color=#800000> del vector <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_azul.gif"
width=15></FONT></FONT><FONT
color=#800000>. </FONT> </TD></TR></TBODY></TABLE></CENTER>
<BR><FONT size=+1> <FONT color=#008080>
Ejemplo:</FONT></FONT> <BR>
<UL>

```



```

<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
  <TBODY>
    <TR>
      <TD>
        <CENTER><IMG
          alt="Gráfica # 13 de vectores y el espacio tridimensional" height=300
          src="..\Pslmagen/vect_graf_13.gif"
width=246></CENTER></TD></TR>
    <TR>
      <TD>
        <CENTER><IMG alt="Vector A" height=20
          src="..\Pslmagen/vector_A_marron.gif" width=15><FONT
            face="Times New Roman,Times"><FONT color=#800000><FONT size=+1> =
            {<I>4, 2,
3</I>}</FONT></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL>
</UL><FONT
color=#000000><FONT size=+1> En términos de componentes, la
suma de vectores se define como sigue:</FONT></FONT> <BR><FONT
color=#000000> </FONT>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
  <TBODY>
    <TR>
      <TD><FONT size=+1><FONT color=#3333ff> </FONT><FONT
        color=#800000>Sean <IMG alt="Vector A" height=20
        src="..\Pslmagen/vector_A_azul.gif" width=15> </FONT><FONT
        color=#0000ff>= <x<SUB>1</SUB>, y<SUB>1</SUB>,
        z<SUB>1</SUB>></FONT><FONT color=#800000> y <IMG alt="Vector B"
        height=20 src="..\Pslmagen/vector_B_azul.gif" width=15>
        </FONT><FONT color=#0000ff>= <x<SUB>2</SUB>, y<SUB>2</SUB>,
        z<SUB>2</SUB>></FONT><FONT color=#800000>, la suma de <IMG
        alt="Vector A" height=20 src="..\Pslmagen/vector_A_marron.gif"
        width=15> y <IMG alt="Vector B" height=20
        src="..\Pslmagen/vector_B_marron.gif" width=15> se define
        como:</FONT></FONT><FONT color=#800000> </FONT>
        <CENTER><IMG alt="Vector A" height=20
        src="..\Pslmagen/vector_A_azul.gif" width=15><FONT
        color=#0000ff><FONT size=+1> + <IMG alt="Vector B" height=20
        src="..\Pslmagen/vector_B_azul.gif" width=15> = <x<SUB>1
        </SUB>+ x<SUB>2</SUB>, y<SUB>1 </SUB>+ y<SUB>2</SUB>, z<SUB>1 </SUB>+
z<SUB>2</SUB>></FONT> </FONT> </CENTER></TD></TR></TBODY></TABLE></CENTER>
<BR><FONT size=+1> <FONT color=#008080>
Ejemplo:</FONT></FONT>
<UL>
<UL>
  <CENTER>
    <TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
      <TBODY>
        <TR>
          <TD>
            <CENTER><IMG
              alt="Gráfica # 14 de vectores y el espacio tridimensional" height=270

```

```

src="../../Pslmagen/vect_graf_14.gif"
width=300></CENTER></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector A" height=20
src="../../Pslmagen/vector_A_marron.gif" width=15><FONT
face="Times New Roman,Times"><FONT color=#800000><FONT size=+1><I>
</I>= { <I>3, 2, 3</I>} <IMG alt="Vector B" height=20
src="../../Pslmagen/vector_B_marron.gif" width=15> = {<I>2,
2, 0</I>}</FONT></FONT></FONT></CENTER></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector C" height=20
src="../../Pslmagen/vector_C_marron.gif" width=15><FONT
face="Times New Roman,Times"><FONT color=#800000><FONT size=+1>
= <IMG alt="Vector A" height=20
src="../../Pslmagen/vector_A_marron.gif" width=15><I>
+ <IMG alt="Vector B" height=20
src="../../Pslmagen/vector_B_marron.gif" width=15><I> =
{<I>5, 4,
3</I>}</FONT></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL
></UL> <FONT
color=#000000> <FONT size=+1>Y la igualdad entre vectores se define entonces de
la siguiente manera:</FONT> </FONT> <BR>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="70%">
<TBODY>
<TR>
<TD><FONT size=+1> <FONT color=#3333ff> </FONT><FONT
color=#800000>Sean <IMG alt="Vector A" height=20
src="../../Pslmagen/vector_A_marron.gif" width=15> =
<a<SUB>1</SUB>, a<SUB>2</SUB>, a<SUB>3</SUB>> y <IMG
alt="Vector B" height=20 src="../../Pslmagen/vector_B_marron.gif"
width=15> = <b<SUB>1</SUB>, b<SUB>2</SUB>, b<SUB>3</SUB>>,
entonces <IMG alt="Vector A" height=20
src="../../Pslmagen/vector_A_azul.gif" width=15> = <IMG
alt="Vector A" height=20 src="../../Pslmagen/vector_B_azul.gif"
width=15> si y sólo si</FONT></FONT><FONT
color=#800000> </FONT>
<CENTER><FONT color=#800000> </FONT><FONT color=#0000ff><FONT
size=+1>a<SUB>1</SUB> </SUB>= b<SUB>1</SUB>, a<SUB>2</SUB> </SUB>= b<SUB>2</SUB>,
a<SUB>3</SUB> </SUB>= b<SUB>3</SUB></FONT></FONT><FONT
color=#800000> </FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
<BR>
<H1>
<HR width="100%">
<I><FONT color=#008080>La magnitud de un vector en términos de sus
componentes.</FONT></I></H1> <BR> <FONT size=+1>Por el
teorema de <FONT color=#008080>Pitágoras</FONT>,</FONT> <BR><FONT
size=+1> </FONT>
<CENTER>
<TABLE bgColor=#993366 cellPadding=10 cellSpacing=0 cols=1 width="80%">
<TBODY>
<TR>
<TD><FONT color=#ffffff><FONT size=+2> </FONT><FONT

```

```

size=+1>Sea <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15> =
<a<SUB>1</SUB>,
a<SUB>2</SUB>, a<SUB>3</SUB>>, entonces </FONT></FONT>
<CENTER><FONT color=#ffffff><B>|<IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15></B><FONT
size=+1>| = ( a<SUB>1</SUB><SUP>2</SUP> + a<SUB>2</SUB><SUP>2</SUP> +
a<SUB>3</SUB><SUP>2</SUP>
)<SUP>1/2</SUP></FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
<BR> <FONT color=#000000><FONT size=+1>Ahora definiremos
otra operación, la multiplicación de un vector por un
escalar. </FONT></FONT> <BR><FONT color=#000000> </FONT>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
<TBODY>
<TR>
<TD> <FONT size=+1><FONT color=#3333ff> </FONT><FONT
color=#800000>Sea <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_marron.gif" width=15> =
<a<SUB>1</SUB>, a<SUB>2</SUB>, a<SUB>3</SUB>> y k un escalar,
entonces definiremos la multiplicación por un escalar como
sigue:</FONT></FONT> <BR><FONT color=#800000><FONT
size=+1> </FONT></FONT>
<CENTER><FONT color=#800000><FONT size=+1>k <IMG alt="Vector A"
height=20 src="..\Pslmagen/vector_A_marron.gif" width=15> =
<k a<SUB>1</SUB>, k a<SUB>2</SUB>, k
a<SUB>3</SUB>> </FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
<BR> <BR><FONT size=+1> <FONT
color=#008080> Ejemplo:</FONT></FONT>
<UL>
<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2 cols=1 width="50%">
<TBODY>
<TR>
<TD>
<CENTER><IMG
alt="Gráfica # 15 de vectores y el espacio tridimensional" height=241
src="..\Pslmagen/vect_graf_15.gif"
width=288></CENTER></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_marron.gif" width=15><FONT
face="Times New Roman, Times" ><FONT color=#800000><FONT size=+1><|>
</|>= {<|>4, 3, 2</|>} <|>k = 1 /
4</|></FONT></FONT></FONT></CENTER></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector C" height=20
src="..\Pslmagen/vector_C_marron.gif" width=15><|><FONT
face="Times New Roman, Times" ><FONT color=#800000><FONT size=+1>=
k <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_marron.gif" width=15> = {1, 3 /
4, 1 /

```


2}</I></CENTER></TD></TR></TBODY></TABLE></CENTER>

 Obsérvese que k es un vector con la misma dirección que si k > 0 y con dirección contraria si k < 0.
 La magnitud de k es |k| |

<CENTER>

<TABLE bgColor=#ffffcc border=0 cellPadding=10 cols=1 width=\"80%\">
<TBODY>

<TR>

<TD>En virtud de lo anterior,
el negativo de es
<CENTER>- =
(-1) </CENTER></TD></TR></TBODY></TABLE></CENTER>

<H1>

<HR width=\"100%\">

<I>La sustracción o resta de vectores.</I></H1>

<CENTER>

<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width=\"80%\">
<TBODY>

<TR>

<TD>Se define la resta de vectores como:
<CENTER> - = + (-) </CENTER></TD></TR></TBODY></TABLE></CENTER>

<H1>

<HR width=\"100%\">

<I>Propiedades:</I></H1> Con la definición de vector en términos de componentes y con las propiedades de los números reales, puede demostrarse el siguiente teorema.

<CENTER>

<TABLE bgColor=#993366 cellPadding=10 cellSpacing=0 cols=1 width=\"80%\">

```

<TBODY>
<TR>
<TD><FONT color=#ffffff><FONT size=+2>Teorema</FONT></FONT>
<BR><FONT color=#ffffff><FONT size=+1>Sean <IMG alt="Vector A"
height=20 src="..\Pslmagen/vector_A_blanco.gif" width=15>
y <IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_blanco.gif" width=15> dos vectores y
k<SUB>1</SUB> y k<SUB>2</SUB> dos escalares.
Entonces:</FONT> </FONT> <BR> <BR> <FONT
color=#ffffff><FONT size=+1>i) <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15> + <IMG
alt="Vector B" height=20 src="..\Pslmagen/vector_B_blanco.gif"
width=15> = <IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_blanco.gif" width=15> + <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_blanco.gif"
width=15>
Ley conmutativa</FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>ii) <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15> + (<IMG
alt="Vector B" height=20 src="..\Pslmagen/vector_B_blanco.gif"
width=15> + <IMG alt="Vector C" height=20
src="..\Pslmagen/vector_C_blanco.gif" width=15>) = (<IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_blanco.gif"
width=15> + <IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_blanco.gif" width=15>) + <IMG
alt="Vector C" height=20 src="..\Pslmagen/vector_C_blanco.gif"
width=15> Ley asociativa</FONT> </FONT>
<BR><FONT color=#ffffff><FONT size=+1>iii) <IMG alt="Vector A"
height=20 src="..\Pslmagen/vector_A_blanco.gif" width=15>
+ <IMG alt="Vector O" height=20
src="..\Pslmagen/vector_O_blanco.gif" width=15> = <IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_blanco.gif"
width=15>
Identidad aditiva</FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>iv) <IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15> + (<IMG
alt="Vector A" height=20 src="..\Pslmagen/vector_A_blanco.gif"
width=15>) = <IMG alt="Vector O" height=20
src="..\Pslmagen/vector_O_blanco.gif"
width=15>
Inverso aditivo</FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>v) k<SUB>1</SUB>(<IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15> + <IMG
alt="Vector B" height=20 src="..\Pslmagen/vector_B_blanco.gif"
width=15>) = k<SUB>1<IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15></SUB> +
k<SUB>1<IMG alt="Vector B" height=20
src="..\Pslmagen/vector_B_blanco.gif"
width=15></SUB></FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>vi) (k<SUB>1</SUB> + k<SUB>2</SUB>) <IMG alt="Vector A"
height=20 src="..\Pslmagen/vector_A_blanco.gif" width=15> =
k<SUB>1<IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif" width=15></SUB> +
k<SUB>2<IMG alt="Vector A" height=20
src="..\Pslmagen/vector_A_blanco.gif"
width=15></SUB></FONT> </FONT> <BR><FONT color=#ffffff><FONT

```

```

size=+1>vii) k<SUB>1</SUB>( k<SUB>2<IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15></SUB>) =
(k<SUB>1</SUB>k<SUB>2</SUB>) <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif"
width=15></FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>viii) 1 <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15> = <IMG
alt="Vector A" height=20 src="\"../Pslmagen/vector_A_blanco.gif"
width=15></FONT> </FONT> <BR><FONT color=#ffffff><FONT
size=+1>ix) 0 <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_blanco.gif" width=15> = <IMG
alt="Vector O" height=20 src="\"../Pslmagen/vector_O_blanco.gif"
width=15></FONT> </FONT></TD></TR></TBODY></TABLE></CENTER>
<H1>
<HR width="100%">
<|><FONT color=#008080>Los vectores unitarios <IMG align=ABSCENTER
alt="Vector i unitario" height=30
src="\"../Pslmagen/vector_i_verde.gif" width=18>, <IMG align=ABSC
alt="Vector j unitario" height=30
src="\"../Pslmagen/vector_j_verde.gif" width=18 ENTER> y <IMG
align=ABSCENTER alt="Vector k unitario" height=30
src="\"../Pslmagen/vector_k_verde.gif" width=18>.</FONT></|></H1><FONT
size=+1>Cualquier vector <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_verde.gif" width=15><B> <IMG
alt="Distinto" height=15 src="\"../Pslmagen/distint_verde.gif"
width=13> <IMG alt="Vector O" height=20
src="\"../Pslmagen/vector_O_verde.gif" width=15></B> dá origen a un
vector<B> <IMG alt="Vector A" height=17
src="\"../Pslmagen/vector_Amu_verde.gif"></B> con la misma dirección
pero de magnitud 1, de la siguiente manera:</FONT> <BR> <BR>
<CENTER>
<TABLE border=0>
  <TBODY>
    <TR>
      <TD></TD>
      <TD></TD>
      <TD>
        <CENTER><FONT color=#008080><FONT size=+1>1</FONT></FONT></CENTER></TD>
      <TD></TD></TR>
    <TR>
      <TD><IMG alt="Vector unitario a" height=17
        src="\"../Pslmagen/vector_Amu_verde.gif" width=15></TD>
      <TD><FONT color=#008080><FONT size=+1>=</FONT></FONT></TD>
      <TD>
        <HR noShade width="100%">
      </TD>
      <TD><IMG align=ABSCENTER alt="Vector A" height=20
        src="\"../Pslmagen/vector_A_verde.gif" width=15></TD></TR>
    <TR>
      <TD></TD>
      <TD></TD>
      <TD><FONT color=#008080>|<IMG align=ABSCENTER alt="Vector A" height=20
        src="\"../Pslmagen/vector_A_verde.gif" width=15>|</FONT></TD>
      <TD></TD></TR></TBODY></TABLE></CENTER>
<P><FONT color=#008080><FONT size=+1>Ejemplo:</FONT></FONT>
<UL>

```



```

<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2>
<TBODY>
<TR>
<TD>
<CENTER><IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_marron.gif" width=15><FONT
color=#800000><FONT size=+1> = {3, 1, 4}
| <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_marron.gif" width=15> | =
(26)<SUP>1/2</SUP></FONT></FONT></CENTER></TD></TR>
<TR>
<TD>
<CENTER><IMG alt="Vector unitario a" height=17
src="\"../Pslmagen/vector_Amu_marron.gif" width=15><FONT
color=#800000><FONT size=+1> = {3/(26)<SUP>1/2</SUP>,
1/(26)<SUP>1/2</SUP>, 4/(26)<SUP>1/2</SUP>}
| <IMG alt="Vector unitario a" height=17
src="\"../Pslmagen/vector_Amu_marron.gif" width=15> | =
1</FONT></FONT></CENTER></TD></TR></TBODY></TABLE></CENTER> </UL></UL><FONT
size=+1> Por las definiciones dadas anteriormente, cualquier
vector</FONT> <BR> <BR>
<CENTER>
<TABLE border=0>
<TBODY>
<TR>
<TD><IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_verde.gif" width=15><FONT
color=#008080> <FONT size=+1>= <a<SUB>1</SUB>, a<SUB>2</SUB>,
a<SUB>3</SUB></FONT></FONT></TD></TR></TBODY></TABLE></CENTER>
<FONT size=+1>se puede escribir en la forma</FONT> <BR> <BR>
<CENTER>
<TABLE bgColor=#ffffcc border=0>
<TBODY>
<TR>
<TD><IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_marron.gif" width=15><FONT
color=#800000> <FONT size=+1>= a<SUB>1</SUB><1, 0, 0> +
a<SUB>2</SUB><0, 1, 0> + a<SUB>3</SUB><0, 0,
1></FONT></FONT></TD></TR></TBODY></TABLE></CENTER>
<CENTER>
<TABLE bgColor=#ffffcc cellPadding=10 cellSpacing=0 cols=1 width="80%">
<TBODY>
<TR>
<TD><FONT color=#800000><FONT
size=+1>Definición:</FONT> </FONT> <BR><FONT
color=#800000><FONT size=+1>Definimos los vectores
unitarios</FONT></FONT><FONT color=#3333ff> </FONT>
<CENTER><IMG alt="Vector i unitario" height=20
src="\"../Pslmagen/vector_i_azul.gif" width=12><FONT
color=#3333ff><FONT size=+1> = <1,0,0></FONT> </FONT></CENTER>
<CENTER><IMG alt="Vector j unitario" height=20
src="\"../Pslmagen/vector_j_azul.gif" width=12><FONT
color=#3333ff><FONT size=+1> = <0,1,0></FONT> </FONT></CENTER>

```

```

<CENTER><IMG alt="Vector k unitario" height=20
src="\"../Pslmagen/vector_k_azul.gif" width=12><FONT
color=#3333ff><FONT size=+1> =
<0,0,1></FONT> </FONT></CENTER></TD></TR></TBODY></TABLE></CENTER>
<BR><FONT size=+1> Entonces, por lo anterior, cualquier
vector <IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_verde.gif" width=15> se puede expresar en
la forma</FONT> <BR> <BR>
<CENTER>
<TABLE border=0>
<TBODY>
<TR>
<TD><IMG alt="Vector A" height=20
src="\"../Pslmagen/vector_A_verde.gif" width=15><FONT
color=#008080><FONT size=+1> = a<SUB>1<IMG align=ABSCENTER
alt="Vector i unitario" height=20
src="\"../Pslmagen/vector_i_verde.gif" width=12></SUB> +
a<SUB>2<IMG align=ABSCENTER alt="Vector j unitario" height=20
src="\"../Pslmagen/vector_j_verde.gif" width=12></SUB> +
a<SUB>3</SUB></FONT></FONT><IMG alt="Vector k unitario" height=20
src="\"../Pslmagen/vector_k_verde.gif"
width=12></TD></TR></TBODY></TABLE></CENTER><FONT size=+1> </FONT>
<BR> <FONT size=+1> A los vectores <FONT
color=#008080>a<SUB>1<IMG alt="Vector i unitario" height=20
src="\"../Pslmagen/vector_i_verde.gif" width=12></SUB></FONT><FONT
color=#000000>, </FONT><FONT color=#008080><SUB> </SUB>a<SUB>2<IMG
alt="Vector j unitario" height=20
src="\"../Pslmagen/vector_j_verde.gif" width=12></SUB></FONT>, <FONT
color=#008080>a<SUB>3<IMG alt="Vector k unitario" height=20
src="\"../Pslmagen/vector_k_verde.gif" width=12></SUB></FONT> se les
llama <FONT color=#008080>vectores componentes</FONT>. </FONT> <BR>
<UL>
<UL>
<CENTER>
<TABLE border=1 cellPadding=10 cellSpacing=2>
<TBODY>
<TR>
<TD>
<CENTER><FONT color=#008080><FONT size=+1>Vectores
unitarios <IMG alt="Vector i unitario" height=20
src="\"../Pslmagen/vector_i_verde.gif"
width=12>, <IMG alt="Vector j unitario" height=20
src="\"../Pslmagen/vector_j_verde.gif"
width=12>, </FONT></FONT><IMG alt="Vector k unitario" height=20
width=12 C="\"../vector_k_verde.gif" SR></CENTER></TD></TR>
<TR>
<TD><IMG alt="Gráfica # 16 de vectores y el espacio tridimensional"
height=288 src="\"../Pslmagen/vect_graf_16.gif"
width=265></TD></TR></TBODY></TABLE></CENTER> </UL></UL></UL> <BR>
<HR width="100%">
<a href="\"../PsInteractividad/resultesp.html">
Composición de Fuerzas (Programa Interactivo)</a>

```

```

');

# -----
#
# Table structure for table 'PREGUNTADET'
#

DROP TABLE IF EXISTS PREGUNTADET;
CREATE TABLE PREGUNTADET (
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Materia_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Evalua_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Pregunta_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Pregunta varchar(120) NOT NULL,
  PRIMARY KEY (Docente_ID, Materia_ID, Evalua_ID, Pregunta_ID)
);

#
# Dumping data for table 'PREGUNTADET'
#

INSERT INTO PREGUNTADET VALUES( '0', '1', '1', '1', 'Pregunta 1');
INSERT INTO PREGUNTADET VALUES( '0', '1', '1', '2', 'Pregunta 2');
INSERT INTO PREGUNTADET VALUES( '0', '1', '1', '3', 'Pregunta 3');
INSERT INTO PREGUNTADET VALUES( '0', '1', '1', '4', 'Pregunta 4');
INSERT INTO PREGUNTADET VALUES( '0', '1', '1', '5', 'Pregunta 5');
INSERT INTO PREGUNTADET VALUES( '0', '2', '4', '6', 'Un escalar es:');
INSERT INTO PREGUNTADET VALUES( '0', '2', '4', '7', 'Un Vector es una cantidad que tiene dos
caracteristicas:');
INSERT INTO PREGUNTADET VALUES( '0', '2', '4', '8', 'La igualdad de vectores afirma que dos
vectores A y B son iguales si tienen la misma:');
INSERT INTO PREGUNTADET VALUES( '0', '2', '4', '9', 'Ejemplos de Escalares son:');
INSERT INTO PREGUNTADET VALUES( '0', '2', '4', '10', 'Ejemplos de Vectores son:');
INSERT INTO PREGUNTADET VALUES( '0', '1', '5', '11', 'p1');
INSERT INTO PREGUNTADET VALUES( '0', '1', '5', '12', 'p2');

# -----
#
# Table structure for table 'QUIZ'
#

DROP TABLE IF EXISTS QUIZ;
CREATE TABLE QUIZ (
  User_ID int(255) DEFAULT '0' NOT NULL,
  Sesion_ID int(255) DEFAULT '0' NOT NULL,
  Pregunta_ID int(255) DEFAULT '0' NOT NULL,
  Porcentaje decimal(10,3) DEFAULT '0.000' NOT NULL,
  Correcto int(1) DEFAULT '0' NOT NULL,
  PRIMARY KEY (User_ID, Sesion_ID, Pregunta_ID)
);

#
# Dumping data for table 'QUIZ'
#

INSERT INTO QUIZ VALUES( '0', '98', '1', '1.000', '1');

```

```

INSERT INTO QUIZ VALUES( '0', '116', '1', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '0', '2', '1.500', '1');
INSERT INTO QUIZ VALUES( '0', '0', '1', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '166', '2', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '166', '1', '0.500', '0');
INSERT INTO QUIZ VALUES( '0', '167', '2', '0.500', '0');
INSERT INTO QUIZ VALUES( '0', '167', '1', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '171', '2', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '171', '1', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '172', '1', '4.000', '1');
INSERT INTO QUIZ VALUES( '0', '172', '2', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '172', '3', '3.000', '1');
INSERT INTO QUIZ VALUES( '0', '172', '4', '2.000', '1');
INSERT INTO QUIZ VALUES( '0', '172', '5', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '173', '5', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '173', '4', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '173', '3', '2.000', '1');
INSERT INTO QUIZ VALUES( '0', '173', '2', '0.500', '0');
INSERT INTO QUIZ VALUES( '0', '173', '1', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '180', '3', '2.000', '1');
INSERT INTO QUIZ VALUES( '0', '179', '2', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '176', '1', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '174', '4', '1.500', '1');
INSERT INTO QUIZ VALUES( '0', '174', '5', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '179', '1', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '180', '2', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '180', '1', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '180', '4', '0.500', '0');
INSERT INTO QUIZ VALUES( '0', '181', '8', '0.500', '1');
INSERT INTO QUIZ VALUES( '0', '181', '10', '0.500', '0');
INSERT INTO QUIZ VALUES( '0', '190', '11', '1.000', '1');
INSERT INTO QUIZ VALUES( '0', '190', '12', '0.500', '1');

```

```

# -----
#
# Table structure for table 'RECORD'
#

```

```

DROP TABLE IF EXISTS RECORD;
CREATE TABLE RECORD (
  User_ID int(255) DEFAULT '0' NOT NULL,
  Sesion_ID int(255) DEFAULT '0' NOT NULL,
  Nota decimal(10,2) DEFAULT '0.00' NOT NULL,
  PRIMARY KEY (User_ID, Sesion_ID)
);

```

```

#
# Dumping data for table 'RECORD'
#

```

```

INSERT INTO RECORD VALUES( '0', '98', '1.00');
INSERT INTO RECORD VALUES( '0', '116', '1.00');
INSERT INTO RECORD VALUES( '0', '0', '0.50');
INSERT INTO RECORD VALUES( '0', '166', '1.00');
INSERT INTO RECORD VALUES( '0', '167', '1.00');
INSERT INTO RECORD VALUES( '0', '171', '1.50');

```



```

INSERT INTO RECORD VALUES( '0', '172', '4.50');
INSERT INTO RECORD VALUES( '0', '173', '3.00');
INSERT INTO RECORD VALUES( '0', '174', '1.50');
INSERT INTO RECORD VALUES( '0', '176', '1.00');
INSERT INTO RECORD VALUES( '0', '179', '1.50');
INSERT INTO RECORD VALUES( '0', '180', '2.00');
INSERT INTO RECORD VALUES( '0', '181', '0.50');
INSERT INTO RECORD VALUES( '0', '190', '1.50');

```

```

# -----
#
# Table structure for table 'RESPUESTADET'
#

```

```

DROP TABLE IF EXISTS RESPUESTADET;
CREATE TABLE RESPUESTADET (
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Materia_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Evalua_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Pregunta_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Respuesta_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Respuesta varchar(120) NOT NULL,
  Correcta int(5) DEFAULT '0' NOT NULL,
  PRIMARY KEY (Docente_ID, Materia_ID, Evalua_ID, Pregunta_ID, Respuesta_ID)
);

```

```

#
# Dumping data for table 'RESPUESTADET'
#

```

```

INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '1', '1', 'Respuesta 1', '1');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '1', '2', 'Respuesta 2', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '2', '3', 'Respuesta 1', '1');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '2', '4', 'Respuesta 2', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '3', '5', 'Respuesta 1', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '3', '6', 'Respuesta 2', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '3', '7', 'Respuesta 3', '1');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '4', '8', 'Respuesta 1', '1');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '4', '9', 'Respuesta 2', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '5', '10', 'Respuesta 1', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '5', '11', 'Respuesta 2', '1');
INSERT INTO RESPUESTADET VALUES( '0', '1', '1', '5', '12', 'Respuesta 3', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '6', '13', 'Una cantidad que solo tiene una
magnitud', '1');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '6', '14', 'Una cantidad que tiene dos
magnitudes', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '6', '15', 'Una magnitud que tiene una
cantidad', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '7', '16', 'Velocidad y Masa', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '7', '17', 'Magnitud y direccion', '1');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '7', '18', 'Dirección y Fuerza', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '8', '19', 'Magnitud y Fuerza', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '8', '20', 'Valor y Dirección', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '8', '21', 'Magnitud y direccion', '1');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '9', '22', 'Velocidad y aceleración', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '9', '23', 'Masa y Temperatura', '1');

```

```

INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '9', '24', 'Fuerza y Desplazamiento', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '10', '25', 'Fuerza y Desplazamiento', '1');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '10', '26', 'Area y longitud', '0');
INSERT INTO RESPUESTADET VALUES( '0', '2', '4', '10', '27', 'Dinero y peso', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '5', '11', '28', 'r1', '1');
INSERT INTO RESPUESTADET VALUES( '0', '1', '5', '11', '29', 'r2', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '5', '12', '30', 'r1', '0');
INSERT INTO RESPUESTADET VALUES( '0', '1', '5', '12', '31', 'r2', '1');

```

```

# -----
#
# Table structure for table 'SESION'
#

```

```

DROP TABLE IF EXISTS SESION;
CREATE TABLE SESION (
  sesion_id int(255) DEFAULT '0' NOT NULL auto_increment,
  estatus char(1) DEFAULT 'I' NOT NULL,
  username varchar(50) NOT NULL,
  PRIMARY KEY (sesion_id)
);

```

```

#
# Dumping data for table 'SESION'
#

```

```

INSERT INTO SESION VALUES( '1', 'A', 'salem');
INSERT INTO SESION VALUES( '2', 'A', 'salem');
INSERT INTO SESION VALUES( '3', 'A', 'salem');
INSERT INTO SESION VALUES( '4', 'A', 'ssalem');
INSERT INTO SESION VALUES( '5', 'A', 'salem');
INSERT INTO SESION VALUES( '6', 'A', 'salem');
INSERT INTO SESION VALUES( '7', 'A', 'salem');
INSERT INTO SESION VALUES( '8', 'A', 'salem');
INSERT INTO SESION VALUES( '9', 'A', 'salem');
INSERT INTO SESION VALUES( '10', 'A', 'salem');
INSERT INTO SESION VALUES( '11', 'A', 'salem');
INSERT INTO SESION VALUES( '12', 'A', 'salem');
INSERT INTO SESION VALUES( '13', 'A', 'salem');
INSERT INTO SESION VALUES( '14', 'A', 'salem');
INSERT INTO SESION VALUES( '15', 'A', 'ssalem');
INSERT INTO SESION VALUES( '16', 'A', 'salem');
INSERT INTO SESION VALUES( '17', 'A', 'salem');
INSERT INTO SESION VALUES( '18', 'A', 'salem');
INSERT INTO SESION VALUES( '19', 'A', 'salem');
INSERT INTO SESION VALUES( '20', 'A', 'salem');
INSERT INTO SESION VALUES( '21', 'A', 'salem');
INSERT INTO SESION VALUES( '22', 'A', 'salem');
INSERT INTO SESION VALUES( '23', 'A', 'salem');
INSERT INTO SESION VALUES( '24', 'A', 'salem');
INSERT INTO SESION VALUES( '25', 'A', 'salem');
INSERT INTO SESION VALUES( '26', 'A', 'salem');
INSERT INTO SESION VALUES( '27', 'A', 'salem');
INSERT INTO SESION VALUES( '28', 'A', 'salem');
INSERT INTO SESION VALUES( '29', 'A', 'salem');
INSERT INTO SESION VALUES( '30', 'A', 'salem');

```

[illegible]

[illegible]

```

INSERT INTO SESION VALUES( '143', 'A', 'salem');
INSERT INTO SESION VALUES( '144', 'A', 'salem');
INSERT INTO SESION VALUES( '145', 'A', 'salem');
INSERT INTO SESION VALUES( '146', 'A', 'salem');
INSERT INTO SESION VALUES( '147', 'A', 'salem');
INSERT INTO SESION VALUES( '148', 'A', 'salem');
INSERT INTO SESION VALUES( '149', 'A', 'ssalem');
INSERT INTO SESION VALUES( '150', 'A', 'salem');
INSERT INTO SESION VALUES( '151', 'A', 'ssalem');
INSERT INTO SESION VALUES( '152', 'A', 'ssalem');
INSERT INTO SESION VALUES( '153', 'A', 'ssalem');
INSERT INTO SESION VALUES( '154', 'A', 'salem');
INSERT INTO SESION VALUES( '155', 'A', 'salem');
INSERT INTO SESION VALUES( '156', 'A', 'salem');
INSERT INTO SESION VALUES( '157', 'A', 'ssalem');
INSERT INTO SESION VALUES( '158', 'A', 'salem');
INSERT INTO SESION VALUES( '159', 'A', 'salem');
INSERT INTO SESION VALUES( '160', 'A', 'salem');
INSERT INTO SESION VALUES( '161', 'A', 'salem');
INSERT INTO SESION VALUES( '162', 'A', 'salem');
INSERT INTO SESION VALUES( '163', 'A', 'ssalem');
INSERT INTO SESION VALUES( '164', 'A', 'salem');
INSERT INTO SESION VALUES( '165', 'A', 'salem');
INSERT INTO SESION VALUES( '166', 'A', 'salem');
INSERT INTO SESION VALUES( '167', 'A', 'salem');
INSERT INTO SESION VALUES( '168', 'A', 'ssalem');
INSERT INTO SESION VALUES( '169', 'A', 'salem');
INSERT INTO SESION VALUES( '170', 'A', 'salem');
INSERT INTO SESION VALUES( '171', 'A', 'ssalem');
INSERT INTO SESION VALUES( '172', 'A', 'salem');
INSERT INTO SESION VALUES( '173', 'A', 'ssalem');
INSERT INTO SESION VALUES( '174', 'A', 'salem');
INSERT INTO SESION VALUES( '175', 'A', 'salem');
INSERT INTO SESION VALUES( '176', 'A', 'salem');
INSERT INTO SESION VALUES( '177', 'A', 'salem');
INSERT INTO SESION VALUES( '178', 'A', 'salem');
INSERT INTO SESION VALUES( '179', 'A', 'salem');
INSERT INTO SESION VALUES( '180', 'A', 'salem');
INSERT INTO SESION VALUES( '181', 'A', 'salem');
INSERT INTO SESION VALUES( '182', 'A', 'salem');
INSERT INTO SESION VALUES( '183', 'A', 'ssalem');
INSERT INTO SESION VALUES( '184', 'A', 'salem');
INSERT INTO SESION VALUES( '185', 'A', 'salem');
INSERT INTO SESION VALUES( '186', 'A', 'salem');
INSERT INTO SESION VALUES( '187', 'A', 'salem');
INSERT INTO SESION VALUES( '188', 'A', 'salem');
INSERT INTO SESION VALUES( '189', 'A', 'salem');
INSERT INTO SESION VALUES( '190', 'A', 'salem');
INSERT INTO SESION VALUES( '191', 'A', 'ssalem');
INSERT INTO SESION VALUES( '192', 'A', 'ssalem');
INSERT INTO SESION VALUES( '193', 'A', 'locotus');
INSERT INTO SESION VALUES( '194', 'A', 'salem');

```

```

# -----
# Table structure for table 'SUBTEMAS'
#

```

```

DROP TABLE IF EXISTS SUBTEMAS;
CREATE TABLE SUBTEMAS (
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Materia_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Tema_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Subtema_ID decimal(10,0) DEFAULT '0' NOT NULL,
  nombresubtema varchar(50) NOT NULL,
  PRIMARY KEY (Docente_ID, Materia_ID, Tema_ID, Subtema_ID)
);

#
# Dumping data for table 'SUBTEMAS'
#

INSERT INTO SUBTEMAS VALUES( '0', '1', '0', '0', 'Primer Subtema');
INSERT INTO SUBTEMAS VALUES( '0', '1', '0', '1', 'Segundo Subtema');
INSERT INTO SUBTEMAS VALUES( '0', '2', '3', '2', 'Ley de Ohm');
INSERT INTO SUBTEMAS VALUES( '0', '2', '2', '3', 'Vectores');
INSERT INTO SUBTEMAS VALUES( '0', '3', '4', '4', 'sub tema 1');
INSERT INTO SUBTEMAS VALUES( '0', '1', '0', '5', 'Subtema 2');

# -----
#
# Table structure for table 'TEMAS'
#

DROP TABLE IF EXISTS TEMAS;
CREATE TABLE TEMAS (
  Docente_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Materia_ID decimal(10,0) DEFAULT '0' NOT NULL,
  Tema_ID decimal(10,0) DEFAULT '0' NOT NULL,
  nombretema varchar(50) NOT NULL,
  PRIMARY KEY (Docente_ID, Materia_ID, Tema_ID)
);

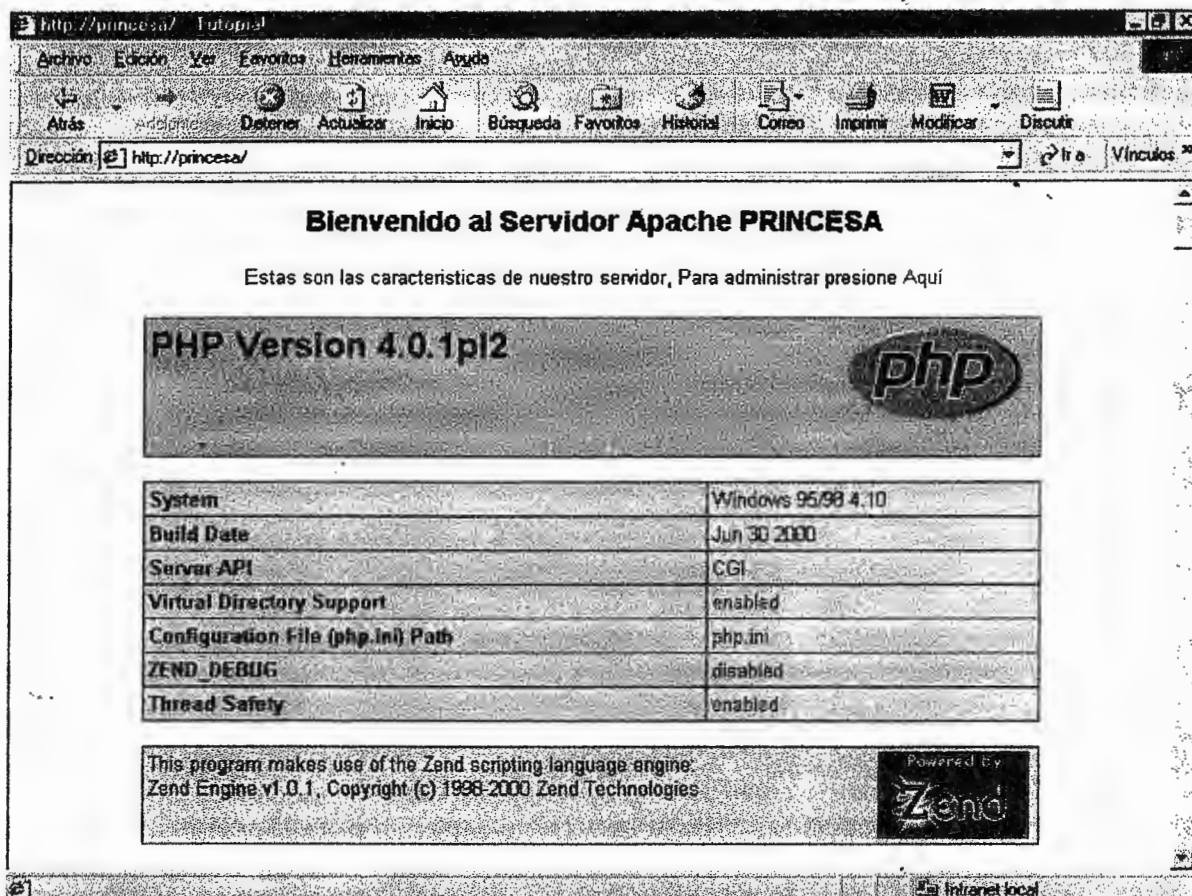
#
# Dumping data for table 'TEMAS'
#

INSERT INTO TEMAS VALUES( '0', '2', '2', 'Mecánica');
INSERT INTO TEMAS VALUES( '0', '1', '0', 'Primer tema');
INSERT INTO TEMAS VALUES( '0', '1', '1', 'Segundo tema');
INSERT INTO TEMAS VALUES( '0', '2', '3', 'Electrodinámica');
INSERT INTO TEMAS VALUES( '0', '3', '4', 'tema 1');

```

1.3 Como utilizar el PhpMyAdmin

La pantalla inicial que permite el ingreso al PhpMyAdmin es la siguiente



La estructura de MySQL

Para cada base de datos que nosotros creamos, MySQL crea un directorio con el nombre que le hemos asignado a la base de datos. Dentro de este directorio, por cada tabla que definamos MySQL va a crear tres archivos: **mitabla.ISD**, **mitabla.ISM**, **mitabla.frm**

El archivo con extensión **ISD**, es el que contiene los datos de nuestra tabla, el **ISM** contiene información acerca de las claves y otros datos que MySQL utiliza para buscar datos en el fichero **ISD**. Y el archivo **frm** contiene la estructura de la propia tabla.

Dado que las bases de datos de MySQL son simples ficheros de un directorio, para realizar copias de seguridad, podremos utilizar las herramientas de compresión que habitualmente usamos en nuestro sistema y luego copiarlo a otro lugar, o simplemente esto último.

Seguridad

Como comentamos anteriormente, todo el sistema de permisos MySQL lo guarda en una base de datos llamada mysql, la cual se componen de cinco tablas: **host**, **user**, **db**, **tables_priv**, **columns_priv**.

La tabla **user** contiene información sobre los usuarios, desde que máquinas pueden acceder a nuestro servidor MySQL, su clave y de sus diferentes permisos. La tabla **host** nos informa sobre que máquinas podran acceder a nuestro sistema, así como a las bases de datos que tendrán acceso y sus diferentes permisos. Finalmente, las tablas **db**, **tables_priv**, **columns_priv** nos proveen de un control individual de las bases de datos, tablas y columnas (campos).

Tabla user

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
User	char(16)	
Password	char(16)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
Reload_priv	enum('N','Y')	N
Shutdown_priv	enum('N','Y')	N
Process_priv	enum('N','Y')	N
File_priv	enum('N','Y')	N
Grant_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla host

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
Db	char(32)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
Grant_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla db

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
Db	char(32)	
User	char(16)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

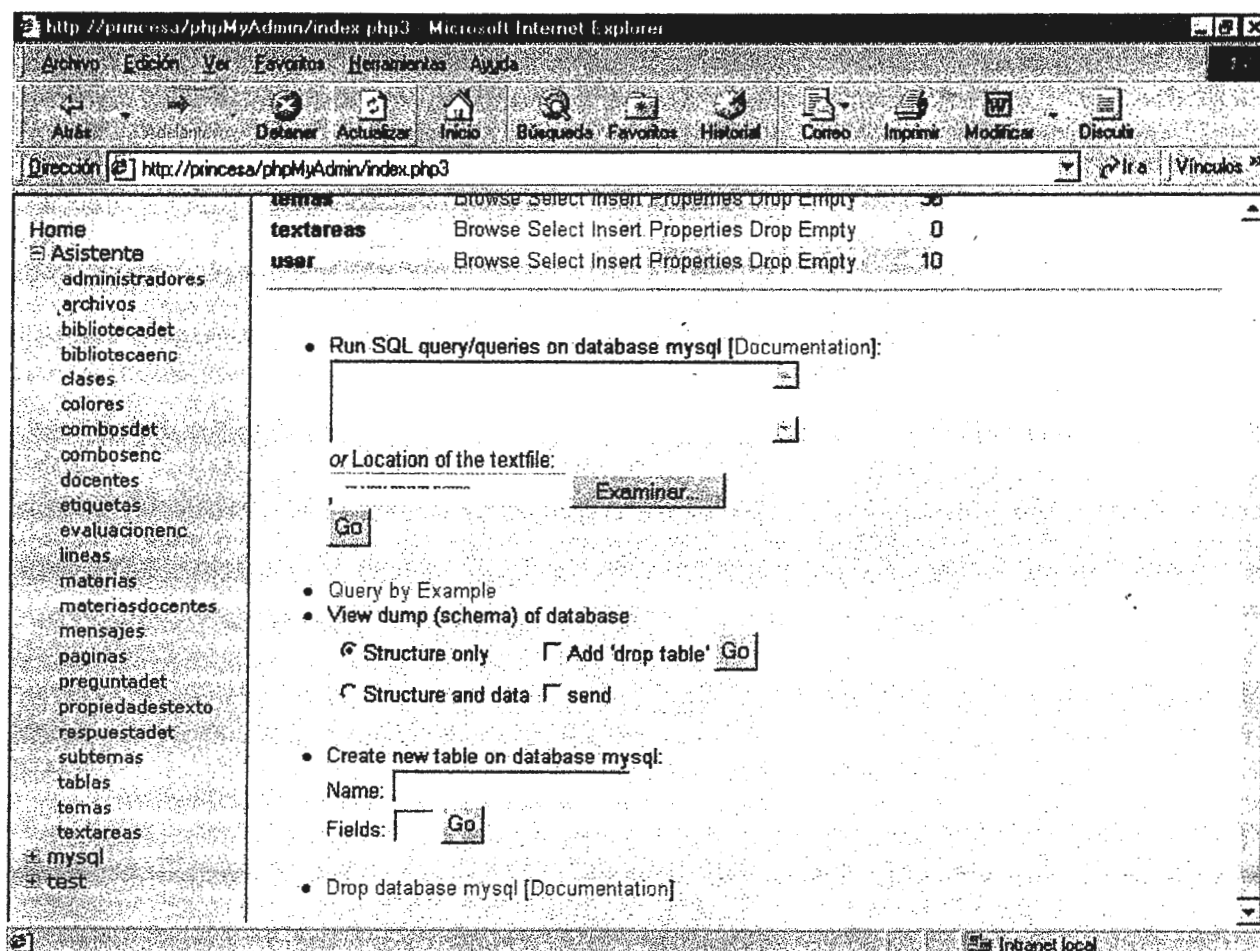
He aquí una breve descripción de los diferentes permisos:

- **Select_priv:** Permite utilizar la sentencia SELECT
- **Insert_priv:** Permite utilizar la sentencia INSERT
- **Update_priv:** Permite utilizar la sentencia UPDATE
- **Delete_priv:** Permite utilizar la sentencia DELETE
- **Create_priv:** Permite utilizar la sentencia CREATE o crear bases de datos
- **Drop_priv:** Permite utilizar la sentencia DROP o eliminar bases de datos

- **Reload_priv:** Permite recargar el sistema mediante *mysqladmin reload*
- **Shutdown_priv:** Permite parar el servidor mediante *mysqladmin shutdown*
- **Process_priv:** Permite manejar procesos del servidor
- **File_priv:** Permite leer y escribir ficheros usando comando como **SELECT INTO OUTFILE** y **LOAD DATA INFILE**
- **Grant_priv:** Permite otorgar permisos a otros usuarios
- **Index_priv:** Permite crear o borrar índices
- **Alter_priv:** Permite utilizar la sentencia **ALTER TABLE**

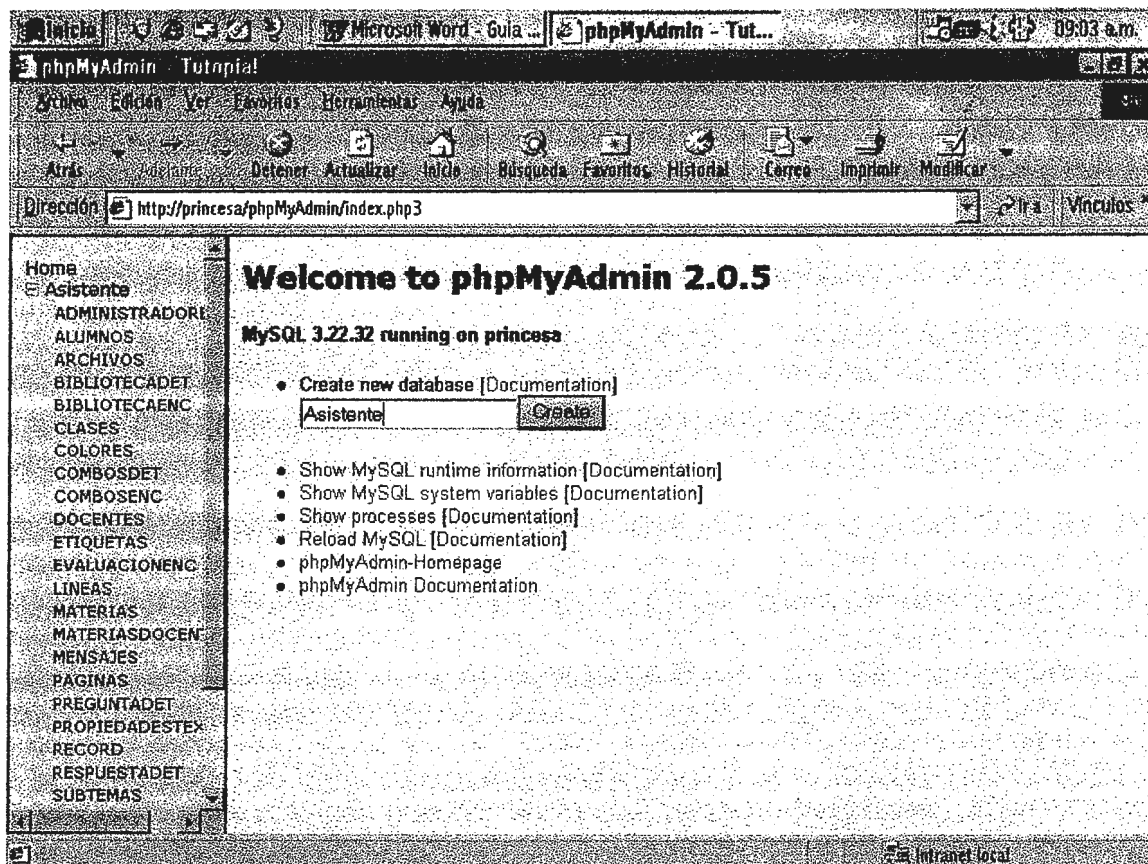
Si dejamos en blanco los campos **user**, **host** o **db**, haremos referencia a cualquier usuario, servidor o base de datos. Conseguiremos el mismo efecto poniendo el símbolo % en el campo.

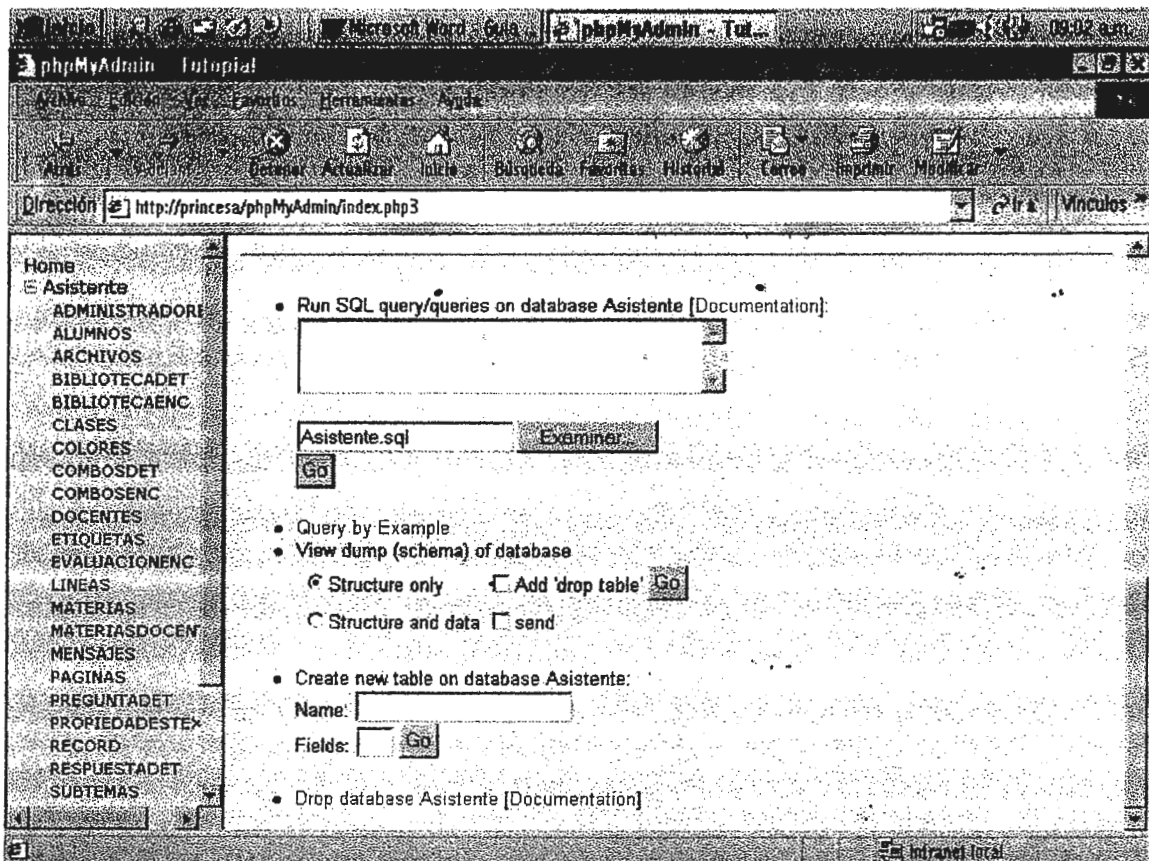
Finalmente para los cambios sean efectuados el Administrador debe digitar la instrucción **FLUSH PRIVILEGES Y LUEGO HACER CLIC EN GO. (ver figura)**



- Crear la base de datos con la herramienta phpMyAdmin. (Esta es la herramienta de administración de bases de datos mysql escrita en php)

- Correr el script Asistente.sql para restablecer la base de datos de nuestra aplicación en el nuevo host, desde la consola phpMyAdmin.





➤ Copiar en htdocs o el directorio de publicación que asigno las carpetas:

- Html
- Php3
- PsImagen
- PsVideo
- PsSonido
- PsInteractividad
- PsVrml

Estas contienen los programas a utilizar para nuestra aplicación.

➤ En el Explorador digitar:

<http://princesa/Html/>

Para ingresar a la página principal de la aplicación. (Recuerde que el nombre del host puede cambiar)

I.4 MySQL Servidor de Bases de Datos

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).

Esta base de datos es considerada (en su propia documentación así lo reseña) como la más rápida y robusta tanto para volúmenes de datos grandes como pequeños (siempre, claro está, comparada con las de su categoría), aunque como veremos más adelante está rapidez es a costa de no implementar ciertos aspectos del SQL.

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.

- 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.
- Tipos de datos: 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.

Seguridad

El sistema de seguridad de MySQL garantiza que cada usuario pueda hacer las cosas que le están permitidas (nada más y nada menos).

El sistema decide los diferentes privilegios dependiendo de "QUÉ USUARIO" conectado a "QUÉ BASE DE DATOS" desde "QUÉ HOST". El sistema de privilegios está basado, como no, en el contenido de 3 tablas, "USER", "HOST" y "DB" de la base de datos "mysql".

1.5 Instalacion de Apahce-Php-Mysql en Linux

Describiremos el proceso de instalación de la base de datos MySQL, de un servidor web Apache con PHP, en una máquina con sistema operativo Linux o Unix.

Lo primero es conseguimos los paquetes necesarios, lo cual lo podemos hacer por medio de las siguientes direcciones de web de los programas utilizados, las cuales son:

-
- **Apache:** <http://www.apache.org/>
 - apache-1.3.x.tar.gz
- **MySQL:** <http://www.mysql.org/>
 - mysql-3_22_22.tar.gz
- **PHP:** <http://www.php.net/>
 - php-3.0.x.tar-gz

Para poder realizar todo el proceso de instalación se debe tener acceso como **root** a la máquina Linux.

Lo primero que debe hacer hacer es un directorio de instalación, aunque lo normal sería que se hiciera en **/usr/local**, **/usr/src**, o bien en **/opt**.

Después de conseguido los paquetes estos deben ser copiados al directorio **root/install**, y luego descomprimirlos primero:

```
cd /usr/local
```

```
tar zxvf /root/install/apache-1.3.x.tar.gz
```

```
tar zxvf /root/install/mysql-3.22.x.tar.gz
```

```
tar zxvf /root/install/php-3.0.x.tar-gz
```

Creamos enlaces sencillos (blandos) a código fuente

```
ln -s /usr/local/apache-1.3.x /usr/local/apache
```

```
ln -s /usr/local/mysql-3.22.x /usr/local/mysql
```

```
ln -s /usr/local/php-3.0.x /usr/local/php
```

Se preparan las fuentes par la compilación de Apache de la siguiente manera:

```
cd /usr/local/apache
```

```
./configure --prefix=/usr/local/apache
```

se compila e instala MySQL, siguiendo los pasos a continuación:

```
cd /usr/local/mysql
```

```
./configure --without-debug --prefix=/usr/local/mysql
```

```
make
```

```
make install
```

```
cp /usr/local/support-files/mysql.server /etc/rc.d/init.d/mysql
```

```
chmod 755 /etc/rc.d/init.d/mysql
```

Se crean las bases del datos del sistema MySQL

```
/usr/local/mysql/bin/mysql_install_db
```

Arrancamos el servidor MySQL

```
/etc/rc.d/init.d/mysql start
```

Se asigna la contraseña del administrador (root) de MySQL

```
/usr/local/mysql/bin/mysqladmin -u root password "clave"
```

Ya con ello se ha concluido con MySQL, ahora se debe compilar PHP como módulo de Apache, siguiendo los pasos a continuación:

```
cd /usr/local/php
```

```
./configure --with-mysql=/usr/local/mysql \
```

```
--with-apache=/usr/local/apache \
```

```
--enable-track-vars
```

```
make
```

```
make install
```

```
#cp php3.ini-dist /usr/local/lib/php3.ini
```

Se compila Apache

```
cd /usr/local/apache
```

```
./configure --prefix=/usr/local/apache \
```

```
--activate-module=src/modules/php3/libphp3.a
```

```
# Si se ha compilado PHP4 entonces utilizar
```

```
#--activate-module=src/modules/php4/libphp4.a
```

```
# quitar los comentarios para habilitar el módulo de proxy
```

```
#--activate-module=src/modules/proxy/libproxy.a
```

```
make
```

```
make install
```

Para definir las extensiones de los scripts PHP, se debe añadir las siguientes líneas en el fichero de configuración de apache (httpd.conf):

```
AddType application/x-httpd-php3 .php3
```

```
AddType application/x-httpd-php3 .php
```

```
AddType application/x-httpd-php3 .phtml
```

Luego de esto solamente falta arrancar el servidor, pero primero se debe hacer una copia del script (copia de seguridad) de arranque en /etc/rc.d/init.d

```
cp /usr/local/apache/bin/apachectl /etc/rc.d/init.d/apache
```

```
/etc/rc.d/init.d/apache start
```

Para comprobar la instalación crea un fichero llamado test.php3 con la siguiente línea:

```
<?php phpinfo() ?>
```

Este fichero debe ser colocado en el directorio de documentos de Apache y llámalo desde el navegador. Si se han seguido todos los pasos bien nos saldrá una página con todas las variables de PHP.

NOTA: Cabe destacar que lo que se ha hecho con cada programa posee su manual por lo que recomendamos leer los manuales para obtener más detalles sobre la instalación de éstas.

1.6 Php3. Componentes de la clase "ini.php3"

Describir las propiedades y métodos de los módulos de Clase (.inc.php3) de php3 que se usan en el proyecto Académicos Web.

La ventaja de crear componentes es crear código reutilizable, de manera que la aplicación se pueda construir en base a éstos; por tanto se creo la Clase Clase MySQL.inc.php3 la cual posee propiedades y métodos definidos para:

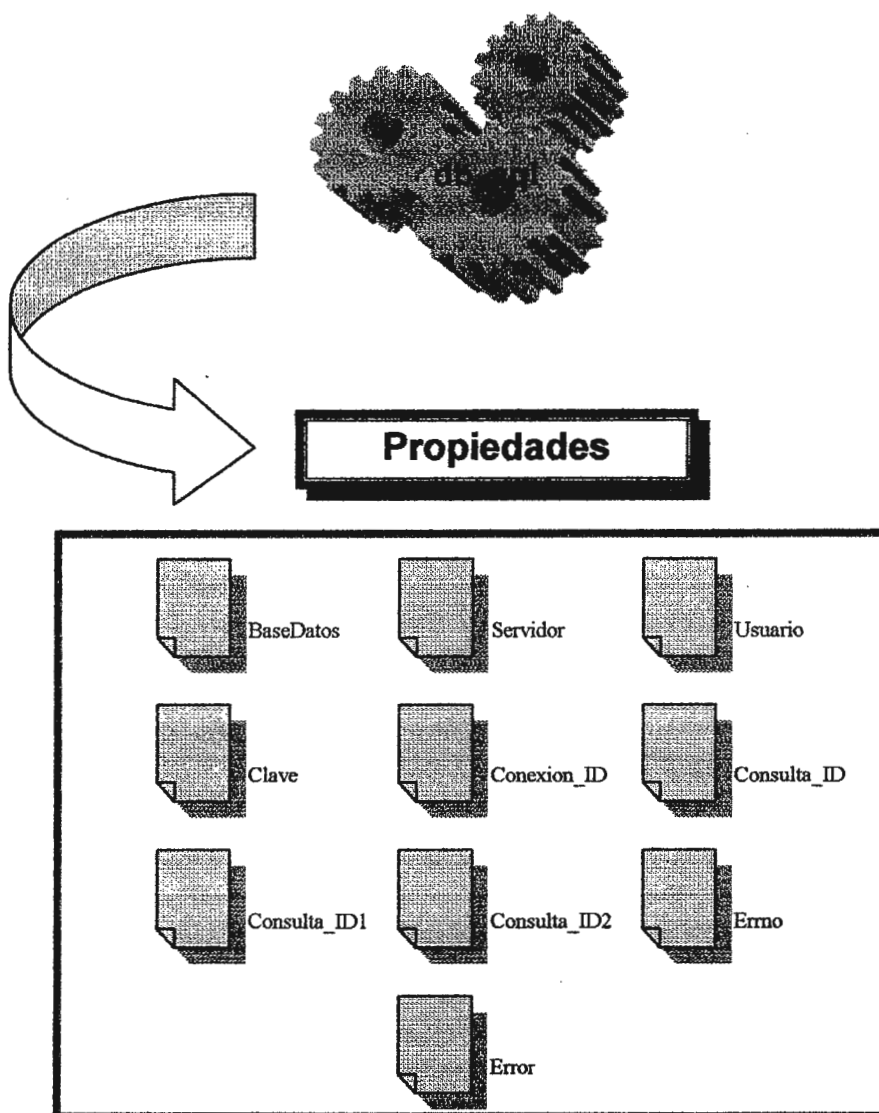
- Crear consultas en cualquier base de datos de MySQL.
- Es capaz de funcionar en cualquier Host o Servidor Web que trabaje con tecnología Php3 o superior.
- Es capaz de construir mantenimientos a partir de una consulta SQL respectiva.
- Es capaz de asociar diferentes tipos de controles HTML a cada campo que refleja una consulta SQL respectiva. Ejemplo: podemos asociar un campo de tipo char(1) a el control HTML ChexBox. (Además de asociarles valores como: 1 chequeado ó 0 no chequeado).
- Es capaz de construir formularios que aniden información padre / hijo registro por registro. Esto implica poder referirnos mediante un vinculo a link a otro lugar del sitio con objeto de verificar el detalle de la información requerida.

- Posee un método especial para correr dos Consultas SQL bajo la misma conexión en el mismo momento. Esto puede utilizarse si se desean a partir de una sola decisión modificar dos tablas o estructuras al mismo tiempo. Ejemplo: Actualizar una tabla y borrar información de otra.
- Es capaz de devolver un valor a partir de una sentencia SQL Respectiva. Ejemplo : Esto se utiliza para mostrar el tiempo para contestar el Quiz en una determinada evaluación.
- Es capaz de reunir la información que se posee una tabla en la base de datos en combo o cuadro de lista con objeto de seleccionar la información apropiada. Ejemplo: Seleccionar una imagen que se desee añadir a la clase en la creación de la misma.
- Es capaz de extraer la información de una consulta SQL y enviarla en un arreglo o matriz con objeto de clasificar la información. (Esto tomo mucha importancia en la creación del Quiz de Ejercicios)

La mayor parte de la aplicación Académicos Web se creo orientado a objetos bajo esta metodología de programación. Por ende basta conocer el modelo de programación utilizado para comprender el manejo de las rutinas estándares de la aplicación. Para visualizar ejemplos completos puede referirse al conjunto de programas fuentes que conforman el módulo del Quiz de Ejercicios, algunos de ellos son: verquiz.php3, resultado.php3, pregunta.php3, respuesta.php3.

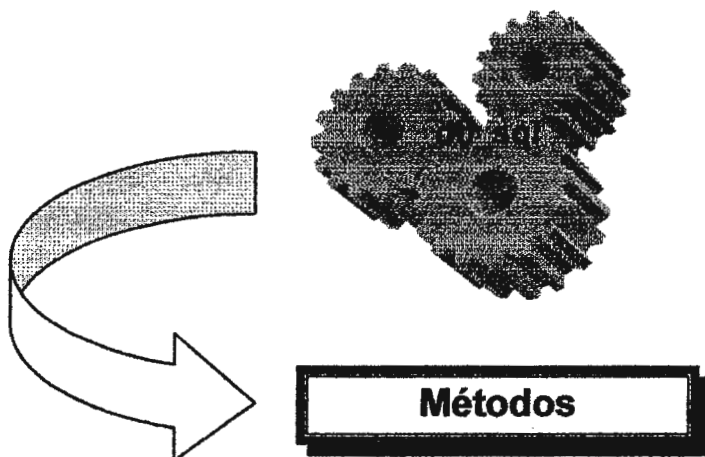
Clase_MySql.inc.php3

Modelo de Objetos Propiedades



Clase_MySql.inc.php3

Modelo de Objetos Metodos



```

class_mysql.inc.php3
├── db_sql
│   ├── db_sql($bd = "", $host = "localhost", $user = "nobody", $pass = "")
│   ├── conectar($bd, $host, $user, $pass)
│   ├── consulta($sql = "")
│   ├── consultadoble($sql1 = "", $sql2 = "")
│   ├── numcampos()
│   ├── numregistros()
│   ├── nombrecampo($numcampo)
│   ├── verconsulta()
│   ├── veropciones()
│   ├── vervalor()
│   ├── verconsultamtto($consulta, $editar, $eliminar, $diferente, $respuesta, $nombrerespuesta)
│   ├── verconsultamttoeditar($editando)
│   ├── verconsultamttoeditartipo($editando, $campo, $largo, $c, $l)
│   ├── verarreglo()
│   ├── verquizpr()
│   ├── sql4arreglo()
│   ├── verconsultalink($consulta1, $nombre1, $consulta2, $nombre2, $link, $diferente)
│   ├── verconsultalinkpfijo($consulta1, $nombre1, $consulta2, $nombre2, $link, $diferente)
│   └── verconsultadatos()

```

PROPIEDADES

BaseDatos

Establece el nombre de la base de datos a conectarse.

Servidor

Asigna el nombre del host o servidor al cual se desea establecer una conexión.

Usuario

Establece el nombre del usuario que hace la conexión.

Clave

Establece la clave asociada al usuario.

Conexión_ID

Mantiene un identificador para la conexión asociada.

Consulta_ID (Consulta_ID1, Consulta_ID2)

Mantiene el identificador de la consulta a la conexión asociada.

Errno

Establece el numero de error generado por la conexión o consulta.

Error

Describe el error generado por la conexión o consulta.

METODOS

db_sql

Es el método constructor de la clase u objeto, se encarga de establecer las propiedades \$db, \$host \$user o \$pass con sus valores iniciales para poder establecer la conexión a un servidor y base de datos respectivos. Sus propiedades privadas son las siguientes:

- \$db : Propiedad privada de la clase para establecer el nombre de la base de datos
- \$host : Propiedad privada de la clase para establecer el servidor disponible para la conexión.
- \$user : Propiedad privada de la clase para establecer el usuario con los privilegios mínimos necesarios para establecer conexión con la base de datos.
- \$pass: Propiedad privada de la clase para establecer la contraseña del usuario.

conectar

Es el método que se utiliza para establecer una conexión con bases de datos MySQL, este nos devuelve un identificador de conexión y nos ayuda establecer las consultas respectivas. Sus propiedades privadas son las siguientes:

- \$db : Propiedad privada de la clase para establecer el nombre de la base de datos
- \$host : Propiedad privada de la clase para establecer el servidor disponible para la conexión.

- **\$user** : Propiedad privada de la clase para establecer el usuario con los privilegios mínimos necesarios para establecer conexión con la base de datos.
- **\$pass**: Propiedad privada de la clase para establecer la contraseña del usuario.

En caso de éxito devuelve el identificador de conexión de lo contrario devuelve cero.

consulta

Es el método que se utiliza para establecer una consulta a una base de datos MySQL, este nos devuelve un identificador de consulta. Su propiedad privada es la siguiente:

- **\$sql** : Propiedad privada de la clase para establecer la sentencia SQL respectiva a ser ejecutada en la base de datos. Puede ser cualquier sentencia que se encuentre dentro del estándar de MySQL.

En caso de éxito devuelve el identificador de consulta de lo contrario devuelve cero.

consultadoble

Es el método que se utiliza para establecer dos consultas simultaneas a una base de datos MySQL, este nos devuelve dos identificadores de consulta. Sus propiedades privadas son las siguientes:

- **\$sql1** : Propiedad privada de la clase para establecer la primera sentencia SQL respectiva a ser ejecutada en la base de datos. Puede ser cualquier sentencia que se encuentre dentro del estándar de MySQL.

- **\$sql2** : Propiedad privada de la clase para establecer la segunda sentencia SQL respectiva a ser ejecutada en la base de datos. Puede ser cualquier sentencia que se encuentre dentro del estándar de MySQL.

En caso de éxito devuelve los identificadores de consulta de lo contrario devuelve cero.

numcampos

Método que devuelve el número de campos de una consulta SQL .

numregistros

Método que devuelve el número de registros de una consulta SQL.

nombrecampo

Método que nos devuelve el nombre de un campo de una consulta SQL. Su propiedad privada es la siguiente:

- **\$numcampo**: Propiedad privada de la clase para establecer el valor según el índice respectivo que toman los campos en una consulta SQL. El primer campo en una consulta SQL corresponde al índice 0.

verconsulta

Es el método que se utiliza para mostrar los resultados de una consulta SQL. Esta es presentada en una tabla en formato HTML.

veropciones

Es el método que se utiliza para llenar con datos de nuestra base, los combos o cuadros de lista a mostrar en una página Web. Los datos dependen de la sentencia SQL que se haya ejecutado anteriormente.

vervalor

Es el método que se utiliza para extraer un determinado valor de la base de datos a partir de una consulta SQL.

verconsultamtto

Es el método que se utiliza para elaborar un menú de mantenimiento, en el cual se pueda consultar información más detallada del registro seleccionado, modificar el registro seleccionado o borrar el registro, a partir de una consulta SQL. Sus propiedades privadas son las siguientes:

- **\$consulta** : Propiedad privada de la clase para establecer el programa php3 para mostrar información más detallada sobre el registro.
- **\$editar** : Propiedad privada de la clase para establecer el programa php3 para modificar el registro seleccionado.
- **\$eliminar** : Propiedad privada de la clase para establecer el programa php3 para eliminar el registro seleccionado.
- **\$diferente** : Propiedad privada de la clase para establecer el valor del índice según el campo de la consulta SQL que nos identifica el registro en la misma y hace la diferencia entre uno y otro registro.
- **\$respuesta** : Propiedad privada de la clase para establecer el programa php3 para dar mantenimiento a las respuestas del registro seleccionado, asumiendo que la consulta este referida a las preguntas de una evaluación.

- **\$nombrerespuesta** : Propiedad privada de la clase para establecer el nombre que se le asociara a la propiedad **\$respuesta**.

verconsultamttoeditar

Es el método que se utiliza para mostrar un formulario para modificar el resultado de una consulta SQL. Su propiedad privada es la siguiente:

- **\$editando** : Propiedad privada de la clase para establecer el programa php3 que ejecutara la sentencia SQL para actualizar los datos en la base.

verconsultamttoeditartipo

Es el método que se utiliza para mostrar un formulario para modificar el resultado de una consulta SQL formateando la salida del campo en la pagina web. Esto Permite asociar diferentes tipos de controles HTML a cada campo que refleja la consulta SQL respectiva.

Sus propiedades privadas son las siguientes:

- **\$editando** : Propiedad privada de la clase para establecer el programa php3 que ejecutara la sentencia SQL para actualizar los datos en la base.
- **\$campo** : Propiedad privada de la clase para establecer el arreglo que contiene los tipos de controles HTML para poder formatear el formulario.
- **\$largo** : Propiedad privada de la clase para establecer el arreglo que contiene el ancho o longitud de los controles HTML para poder formatear el formulario.
- **\$c, \$l** : Propiedad privada de la clase para establecer el numero de campos a mostrar.

verarreglo

Es el método que se utiliza para elaborar un arreglo conforme a una consulta SQL ejecutada.

verquizpr

Es el método que se utiliza para mostrar los resultados de dos consultas SQL. Estas son presentadas en una tabla en formato HTML. Su nombre se deriva que en el inicio se creo con fines de desarrollar el quiz de ejercicios de Académicos Web.

sqlAarreglo

Es el método que se utiliza para trasladar los resultados de una consulta SQL a un arreglo.

verconsultalink

Es el método que se utiliza para mostrar en una tabla con formato HTML los resultados de una consulta con objeto de enlazar o anidar otras paginas respectivas del sitio. Sus propiedades privadas son las siguientes:

- **\$consulta1**: Propiedad privada de la clase para establecer el primer programa php3 a vincular con la pagina activa en el momento.
- **\$nombre1** : Propiedad privada de la clase para establecer el nombre que se le asociara a la propiedad \$consulta1.
- **\$consulta2**: Propiedad privada de la clase para establecer el segundo programa php3 a vincular con la pagina activa en el momento.
- **\$nombre2** : Propiedad privada de la clase para establecer el nombre que se le asociara a la propiedad \$consulta2.

- **\$link**: Propiedad privada de la clase para establecer el número de vínculos a establecer por el método.
- **\$diferente** : Propiedad privada de la clase para establecer el valor del índice según el campo de la consulta SQL que nos identifica el registro en la misma y hace la diferencia entre uno y otro registro.

Las propiedades privadas *consulta1* y *consulta2* deben obligatoriamente llamar a programas *php3* con parámetros.

verconsultalinkpfijo

Funciona de forma similar a *verconsultalink* con la diferencia que las propiedades privadas *consulta1* y *consulta2* deben obligatoriamente llamar a programas *php3* sin parámetros. Sus propiedades privadas son las siguientes:

- **\$consulta1**: Propiedad privada de la clase para establecer el primer programa *php3* a vincular con la pagina activa en el momento.
- **\$nombre1** : Propiedad privada de la clase para establecer el nombre que se le asociara a la propiedad **\$consulta1**.
- **\$consulta2**: Propiedad privada de la clase para establecer el segundo programa *php3* a vincular con la pagina activa en el momento.
- **\$nombre2** : Propiedad privada de la clase para establecer el nombre que se le asociara a la propiedad **\$consulta2**.
- **\$link**: Propiedad privada de la clase para establecer el número de vínculos a establecer por el método.
- **\$diferente** : Propiedad privada de la clase para establecer el valor del índice según el campo de la consulta SQL que nos identifica el registro en la misma y hace la diferencia entre uno y otro registro.

verconsultadatos

Es el método que se utiliza para mostrar los resultados de una consulta SQL. Esta es presentada en una tabla en formato HTML con dos colores intercalados por fila.

1.6.1 Clase_MySql.inc.php3

Código Fuente

```
<?php
```

```
class db_sql {
```

```
/* variables de conexión */
```

```
var $BaseDatos;
```

```
var $Servidor;
```

```
var $Usuario;
```

```
var $Clave;
```

```
/* identificador de conexión y consulta */
```

```
var $Conexion_ID = 0;
```

```
var $Consulta_ID = 0;
```

```
/* número de error y texto error */
```

```
var $Errno = 0;
```

```
var $Error = "";
```

```
/* Método Constructor: Cada vez que creemos una variable
```

```
de esta clase, se ejecutará esta función */
```

```
function db_sql($bd = "", $host = "localhost", $user = "nobody", $pass = "") {
```

```
$this->BaseDatos = $bd;
```

```
$this->Servidor = $host;
```

```
$this->Usuario = $user;
```

```
$this->Clave = $pass;
```

```
}
```

```
/*Conexión a la base de datos*/
```

```
function conectar($bd, $host, $user, $pass){
```

```
if ($bd != "") $this->BaseDatos = $bd;
```

```
if ($host != "") $this->Servidor = $host;
```

```
if ($user != "") $this->Usuario = $user;
```

```
if ($pass != "") $this->Clave = $pass;
```

```
// Conectamos al servidor
```

```
$this->Conexion_ID = mysql_connect($this->Servidor, $this->Usuario, $this->Clave);
```

```
if (!$this->Conexion_ID) {
```

```
    $this->Error = "Ha fallado la conexión.";
```

```
    echo $this->Error;
```

```
    return 0;
```

```
}
```

```
//seleccionamos la base de datos
```

```
if (!mysql_select_db($this->BaseDatos, $this->Conexion_ID)) {
```

```
    $this->Error = "Imposible abrir ".$this->BaseDatos ;
```

```
    echo $this->Error;
```

```
    return 0;
```

```
}
```

```
/* Si hemos tenido éxito conectando devuelve
```

```
el identificador de la conexión, sino devuelve 0 */
```

```
return $this->Conexion_ID;
```

```
}
```

```
/* Ejecuta un consulta */
```

```
function consulta($sql = ""){
```

```
if ($sql == "") {
```

```
$this->Error = "No ha especificado una consulta SQL";
```

```
echo $this->Error;
```

```
return 0;
```

```
}
```

```
//ejecutamos la consulta
```

```
$this->Consulta_ID = mysql_query($sql, $this->Conexion_ID);
```

```
if (!$this->Consulta_ID) {
```

```
$this->Errno = mysql_errno();
```

```
$this->Error = mysql_error();
```

```
echo $this->Errno;
```

```
echo "<br>";
```

```
echo $this->Error;
```

```
}
```

```
/* Si hemos tenido éxito en la consulta devuelve
```

```
el identificador de la conexión, sino devuelve 0 */
```

```
return $this->Consulta_ID;
```

```
}
```

```
/* Ejecuta un consulta */
```

```
function consultadoble($sql1 = "", $sql2 = ""){
```

```
if ($sql1 == "" or $sql2 == "") {
```

```
$this->Error = "No ha especificado una consulta SQL";
```

```
return 0;
```

```
}
```

```
//ejecutamos las consulta
```

```
$this->Consulta_ID1 = mysql_query($sql1, $this->Conexion_ID);
```

```
if (!$this->Consulta_ID1) {
```



```
$this->Errno = mysql_errno();
```

```
$this->Error = mysql_error();
```

```
}
```

```
$this->Consulta_ID2 = mysql_query($sql2, $this->Conexion_ID);
```

```
if (!$this->Consulta_ID2) {
```

```
$this->Errno = mysql_errno();
```

```
$this->Error = mysql_error();
```

```
}
```

```
/* Si hemos tenido éxito en la consulta devuelve
```

```
el identificador de la conexión, sino devuelve 0 */
```

```
return $this->Consulta_ID1;
```

```
return $this->Consulta_ID2;
```

```
}
```

```
/* Devuelve el número de campos de una consulta */
```

```
function numcampos() {
```

```
return mysql_num_fields($this->Consulta_ID);
```

```
}
```

```
/* Devuelve el número de registros de una consulta */
```

```
function numregistros(){
```

```
return mysql_num_rows($this->Consulta_ID);
```

```
}
```

```
/* Devuelve el nombre de un campo de una consulta */
```

```
function nombrecampo($numcampo) {
```

```
return mysql_field_name($this->Consulta_ID, $numcampo);
```

```
}
```

```
/* Muestra los datos de una consulta */
```

```
function verconsulta() {
```

```
echo "<table border=1>\n";
```

```
// mostramos los nombres de los campos
```

```
for ($i = 0; $i < $this->numcampos(); $i++){

    echo "<td><b>".$this->nombrecampo($i)."</b></td>\n";

}

echo "</tr>\n";

// mostrarmos los registros

while ($row = mysql_fetch_row($this->Consulta_ID)) {

    echo "<tr> \n";

    for ($i = 0; $i < $this->numcampos(); $i++){

        echo "<td>".$row[$i]."</td>\n";

    }

    echo "</tr>\n";

}

}

/* Muestra las opciones de un combo para una consulta */

function veropciones() {

    // mostrarmos los registros
```

```
while ($row = mysql_fetch_row($this->Consulta_ID))
{

for ($i = 0; $i < $this->numcampos(); $i++)
{echo '<Option value="'. $row[$i].'">' . $row[$i=$i+1]. '</Option>';}

}

}
```

/ Muestra un determinado valor */*

```
function vervalor() {
```

```
// mostrarmos los registros
```

```
while ($row = mysql_fetch_row($this->Consulta_ID))
{

for ($i = 0; $i < $this->numcampos(); $i++)
{return $row[$i];}

}

}
```

/ Muestra los datos de una consulta para mtto*/*

```
function
verconsultamtto($consulta,$editar,$eliminar,$diferente,$respuesta,$nombrerespue
sta){

if ($diferente=="")
{
    $dif = "null";
}

echo "<table cellpadding=10 cellspacing=10>\n";

// mostramos los nombres de los campos

for ($i = 0; $i < $this->numcampos(); $i++){

echo "<td bgcolor=#9B52F5><b>".$this->nombrecampo($i)."</b></td>\n";

}

// Encabezados de Mtto

echo "<td></td>\n";
echo "<td></td>\n";
echo "<td></td>\n";

echo "</tr>\n";

// mostramos los registros

$color = 1;
```

```
while ($row = mysql_fetch_row($this->Consulta_ID)) {
```

```
    echo "<tr> \n";
```

```
    if (($color%2)==0)
```

```
    {
```

```
        $nuevocolor = "#BB8FF1";
```

```
    }
```

```
    else
```

```
    {
```

```
        $nuevocolor = "#C8C4F2";
```

```
    }
```

```
    for ($i = 0; $i < $this->numcampos(); $i++){
```

```
        if ($diferente == $i)
```

```
        {
```

```
            $dif = $row[$i];
```

```
        }
```

```
        echo "<td bgcolor='$nuevocolor'>".$row[$i]."</td>\n";
```

```
    }
```

```
// Incorporar Mantenimientos o anidamientos de hijos
```

```
If ($respuesta==" Or $nombrerespuesta=="")
```

```
{
```

```
    echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$consulta."&dif=$dif"> Ver  
</a> </td>\n";
```

```

echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$editar."&dif=$dif">  Editar
</a> </td>\n";
echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$eliminar."&dif=$dif">
Eliminar </a> </td>\n";
}
else
{
echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$consulta."&dif=$dif"> Ver
</a> </td>\n";
echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$editar."&dif=$dif">  Editar
</a> </td>\n";
echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$eliminar."&dif=$dif">
Eliminar      </a> </td>\n";
echo "<td bgcolor='$nuevocolor'> <a
href='../Php3/".$respuesta."&dif=$dif">".$nombrerespuesta." </a> </td>\n";
}

echo "</tr>\n";

$color = $color + 1;

}

}

```



```
/* Muestra los datos de una consulta para editarlos*/
```

```
function verconsultamttoeditar($editando){
```

```
echo "<table cellpadding=10 cellspacing=10>\n";
```

```
// mostramos los nombres de los campos
```

```
for ($i = 0; $i < $this->numcampos(); $i++){
```

```
echo "<td bgcolor=#9B52F5><b>".$this->nombrecampo($i)."</b></td>\n";
```

```
}
```

```
echo "</tr>\n";
```

```
// mostrarmos los registros
```

```
$color = 1;
```

```
echo "<center>";
```

```
echo "<form action='".$editando."' method='post'>";
```

```
$con = 0;
```

```
while ($row = mysql_fetch_row($this->Consulta_ID)) {
```

```
echo "<tr> \n";
```

```
if (($color%2)==0)
```

```
{
```

```
$nuevocolor = "#BB8FF1";
}
else
{
    $nuevocolor = "#C8C4F2";
}

for ($i = 0; $i < $this->numcampos(); $i++){

    $con=$con+1;

    echo "<td bgcolor='$nuevocolor'> <input type='text' name='c'.".$con."
value='". $row[$i]."' size=50></td>\n";

}

echo "</tr>\n";

$color = $color + 1;
}

echo "<tr><td bgcolor='$nuevocolor' colspan='". $con."'">";
echo "<center><input type='submit' name='grabar' value='Actualizar...'></center>";
echo "</td></tr>";
echo "</form>";
echo "</center>";

}
```

/* Muestra los datos de una consulta para editarlos con diferente tipo por campo*/

```
function verconsultamttoeditartipo($editando,$campo,$largo,$c,$l){

echo "<table cellpadding=10 cellspacing=10>\n";

// mostramos los nombres de los campos

for ($i = 0; $i < $this->numcampos(); $i++){

echo "<td bgcolor=#9B52F5><b>".$this->nombrecampo($i)."</b></td>\n";

}

echo "</tr>\n";

// mostramos los registros

$color = 1;

echo "<center>";
echo "<form action='".$editando."' method='post'>";

$con = 0;

$le = 0;

while ($row = mysql_fetch_row($this->Consulta_ID)) {

echo "<tr> \n";
```

```

if (($color%2)==0)
{
    $nuevocolor = "#BB8FF1";
}
else
{
    $nuevocolor = "#C8C4F2";
}

for ($i = 0; $i < $this->numcampos(); $i++){

    $con=$con+1;

    if ($campo[$sele]=="checkbox" Or $campo[$sele]=="radio")
    {

        if ($row[$i]==1)
            {$chequeado = "checked";}
        else
            {$chequeado = "onClick='if(this.checked) { this.value=1 }'";}

        echo "<td bgcolor='$nuevocolor'> <input type='\".$campo[$sele].\"' name='c\".$con.\"'
size='\".$largo[$sele].\"' \".$chequeado.\" value='\".$row[$i].\"></td>\n";
    }
    else
    {
        echo "<td bgcolor='$nuevocolor'> <input type='\".$campo[$sele].\"' name='c\".$con.\"'
value='\".$row[$i].\"' size='\".$largo[$sele].\"></td>\n";
    }
}

```

```
$Sele = $Sele + 1;
```

```
}
```

```
echo "</tr>\n";
```

```
$color = $color + 1;
```

```
}
```

```
echo "<tr><td bgcolor='$nuevocolor' colspan='".$con.">";
```

```
echo "<center><input type='submit' name='grabar' value='Actualizar...'></center>";
```

```
echo "</td></tr>";
```

```
echo "</form>";
```

```
echo "</center>";
```

```
}
```

/* Elabora un arreglo conforme a una consulta */

```
function verarreglo() {
```

```
// mostrarmos los registros
```

```
while ($row = mysql_fetch_row($this->Consulta_ID))
```

```
{
```

```
for ($i = 0; $i < $this->numcampos(); $i++)
```

```
{ $arreglo[$row[$i]] = $row[$i+$i+1]; }
```

```
}
```

```
// Retorna el arreglo;
```

```
return $arreglo;
```

```
}
```

```
// Ver Quiz
```

```
function verquizpr() {
```

```
echo "<table border=1>\n";
```

```
// mostrarmos los registros de la consulta 1
```

```
while ($row = mysql_fetch_row($this->Consulta_ID1)) {
```

```
echo "<tr> \n";
```

```
for ($i = 0; $i < mysql_num_fields($this->Consulta_ID1); $i++){
```

```
echo "<td>". $row[$i]. "</td>\n";
```

```
}
```

```
echo "</tr>\n";
```

```
}
```

```
// mostrarmos los registros de la consulta 2
```

```
while ($row = mysql_fetch_row($this->Consulta_ID2)) {

echo "<tr> \n";

    for ($i = 0; $i < mysql_num_fields($this->Consulta_ID2); $i++){

        echo "<td>".$row[$i]."</td>\n";

    }

    echo "</tr>\n";

}

echo "</table>";

}

function sqlAarreglo(){
    settype($p,"array");

    // mostrarmos los registros

    while ($row = mysql_fetch_row($this->Consulta_ID))
    {

        for ($i = 0; $i < $this->numcampos(); $i++)
        {
            $p[$row[$i]][$row[$i+1]]=$row[$i+2];
            $i=$i+2;
        }
    }
}
```



```
}
```

```
}
```

```
return $p;
```

```
}
```

/* Muestra los datos de una consulta con objeto de enlazar otras pantallas*/

```
function
```

```
verconsultalink($consulta1,$nombre1,$consulta2,$nombre2,$link,$diferente){
```

```
if ($diferente=="")
```

```
{
```

```
    $dif = "null";
```

```
}
```

```
if ($link!="")
```

```
{
```

```
    $link = "S";
```

```
}
```

```
echo "<table cellpadding=10 cellspacing=10>\n";
```

// mostramos los nombres de los campos

```
for ($i = 0; $i < $this->numcampos(); $i++){
```

```
echo "<td bgcolor=#9B52F5><b>".$this->nombrecampo($i)."</b></td>\n";
```

```
}
```

// Encabezados de los links

```
echo "<td></td>\n";
```

```
echo "<td></td>\n";
```

```
echo "</tr>\n";
```

// mostrarmos los registros

```
$color = 1;
```

```
while ($row = mysql_fetch_row($this->Consulta_ID)) {
```

```
echo "<tr> \n";
```

```
if (($color%2)==0)
```

```
{
```

```
    $nuevocolor = "#BB8FF1";
```

```
}
```

```
else
```

```
{
```

```
    $nuevocolor = "#C8C4F2";
```

```
}
```

```
for ($i = 0; $i < $this->numcampos(); $i++){
```

```
    if ($diferente == $i)
```

```
    {
```

```
$dif = $row[$i];  
}
```

```
echo "<td bgcolor='$nuevocolor'>",$row[$i]."</td>\n";  
  
}
```

// Incorporar links

```
if ($link=="")  
{  
    echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$consulta1."&dif=$dif">  
    ".$nombre1." </a> </td>\n";  
}  
else  
{  
    echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$consulta1."&dif=$dif">  
    ".$nombre1." </a> </td>\n";  
    echo "<td bgcolor='$nuevocolor'> <a href='../Php3/".$consulta2."&dif=$dif">  
    ".$nombre2." </a> </td>\n";  
}
```

```
echo "</tr>\n";
```

```
$color = $color + 1;
```

```
}
```

```
}
```

/* Muestra los datos de una consulta con objeto de enlazar otras pantallas*/

function

verconsultalinkpfijo(\$consulta1,\$nombre1,\$consulta2,\$nombre2,\$link,\$diferente){

if (\$diferente=="")

{

\$dif = "null";

}

if (\$link!="")

{

\$link = "S";

}

echo "<table cellpadding=10 cellspacing=10>\n";

// mostramos los nombres de los campos

for (\$i = 0; \$i < \$this->numcampos(); \$i++){

echo "<td bgcolor=#9B52F5>".\$this->nombrecampo(\$i)."</td>\n";

}

// Encabezados de los links

```
echo "<td></td>\n";
```

```
echo "<td></td>\n";
```

```
echo "</tr>\n";
```

// mostrarmos los registros

```
$color = 1;
```

```
while ($row = mysql_fetch_row($this->Consulta_ID)) {
```

```
echo "<tr> \n";
```

```
if (($color%2)==0)
```

```
{
```

```
    $nuevocolor = "#BB8FF1";
```

```
}
```

```
else
```

```
{
```

```
    $nuevocolor = "#C8C4F2";
```

```
}
```

```
for ($i = 0; $i < $this->numcampos(); $i++){
```

```
    if ($diferente == $i)
```

```
    {
```

```
        $dif = $row[$i];
```

```
    }
```

```
echo "<td bgcolor='$nuevocolor'>".$row[$i]."</td>\n";
```

```
}
```

// Incorporar links

```
if ($link=="")
```

```
{
```

```
    echo "<td bgcolor='$nuevocolor'> <a href='../Php3/'. $consulta1."?dif=$dif">
    ".$nombre1." </a> </td>\n";
```

```
}
```

```
else
```

```
{
```

```
    echo "<td bgcolor='$nuevocolor'> <a href='../Php3/'. $consulta1."?dif=$dif">
    ".$nombre1." </a> </td>\n";
```

```
    echo "<td bgcolor='$nuevocolor'> <a href='../Php3/'. $consulta2."?dif=$dif">
    ".$nombre2." </a> </td>\n";
```

```
}
```

```
echo "</tr>\n";
```

```
$color = $color + 1;
```

```
}
```

```
}
```

/* Muestra los datos de una consulta con objeto de enlazar otras pantallas*/

```
function verconsultadatos(){
```

```
echo "<table cellpadding=10 cellspacing=10>\n";
```

```
// mostramos los nombres de los campos
```

```
for ($i = 0; $i < $this->numcampos(); $i++){
```

```
echo "<td bgcolor=#9B52F5><b>".$this->nombrecampo($i)."</b></td>\n";
```

```
}
```

```
echo "</tr>\n";
```

```
// mostramos los registros
```

```
$color = 1;
```

```
while ($row = mysql_fetch_row($this->Consulta_ID)) {
```

```
echo "<tr> \n";
```

```
if (($color%2)==0)
```

```
{
```

```
    $nuevocolor = "#BB8FF1";
```

```
}
```

```
else
```

```
{
```

```
    $nuevocolor = "#C8C4F2";
```



```
}
```

```
for ($i = 0; $i < $this->numcampos(); $i++){
```

```
echo "<td bgcolor='$nuevocolor'>".$row[$i]."</td>\n";
```

```
}
```

```
echo "</tr>\n";
```

```
$color = $color + 1;
```

```
}
```

```
}
```

```
} //fin de la Clase db_sql
```

```
?>
```

TABLA 1
LISTA POR GRUPO DE PROGRAMAS

Acceso	FTP	Subtema	Temas	Editor	Clases	Paginas
onectardb.php3	copiar.php3	subtema.php3	tema.php3	Editor_PHP3.php3	actualizarclase.php3	adicionarpagina.php3
gresardocentes.php	subirarchivo.php3	agregartemasubtema.php3	borrartema.php3	Editor_Mod_PHP3.php3	eliminarclase.php3	buscarclasepagina_.php3
enu.php3		borrarsubtema.php3	buscarmateria.php3	Editor	borrarclase.php3	buscarpagina.php3
esconectardb.php3		buscarmateriatema.php3	eliminartema.php3	Editor_PHP3.php3	gclase.php3	buscarclasepagina.php3
gresaralumnos.php3		eliminarsubtema.php3	gtema.php3	Editor_Mod_PHP3.php3	ingresarclase.php3	pbuscarpagina.php3
cceso		gsubtema.php3	ingresartema.php3		iclude.php3	buscarclasepagina_.php3
onectardb.php3		ingresarsubtema.php3	itema.php3		modificarclase.php3	buscarclasepagina_.php3
gresardocentes.php3		isubtema.php3	modificartema.php3		clase.php3	mostrarpagina.php3
enu.php3		modificarsubtema.php3	pbuscarmateria.php3		buscarmateriatemasubtema.php3	pbuscardocente.php3
esconectardb.php3		pbuscarmateriatema.php3	actualizartema.php3		pbuscarmateriatemasubtema.php3	
gresaralumnos.php3		actualizarsubtema.php3			buscarmateriatemasubtema.php3	
					agregartemasubtema.php3	

LISTA POR GRUPO DE PROGRAMAS

QUIZ DE EJERCICIOS	QUIZ DE EJERCICIOS
clase_mysql.inc.php3	mtto_del_respuesta.php3
evaluacion.php3	Autenticacion.inc.php3
ver.php3	index.php3
mtto_add_evaluacion.php3	vernota.php3
mtto_viw_evaluacion.php3	verquiz.php3
editar.php3	resultado.php3
mtto_edt_evaluacion.php3	autoevaluacion.php3
mtto_del_evaluacion.php3	comentario.php3
pregunta.php3	mtto_add_comentario.php3
mtto_add_pregunta.php3	vermensaje.php3
verp.php3	mtto_viw_mensaje.php3
editarp.php3	Escoger_Clase.php3
mtto_edt_pregunta.php3	modificarpagina.php3
mtto_del_pregunta.php3	buscarmateriaevaluaciona.php3
mtto_viw_pregunta.php3	buscarmateriaevaluaciond.php3
respuesta.php3	mtto_del_mensaje.php3
mtto_add_respuesta.php3	buscardocenteevaluaciona.php3
	seleccionedocentecomentario.php3

Chapter 2

JavaScript: The Language

CONTENTS

- JavaScript Syntax
 - Variables and Values
 - Implicit Data Types in JavaScript
 - Type Conversion
 - Statements and Operators
 - The Structure of JavaScript Statements
 - Operators
 - Comments in JavaScript Code
 - Control Structures
 - The if Statement
 - The while Statement
 - The for Statement
 - Functions and Objects
 - Functions
 - Objects
-

There are many, many different computer programming languages in use today. Each has its own set of special features, which are highly praised by its fans and vigorously panned by its detractors. If you have worked in more than one language then you are aware that there is a continuum of language styles, ranging from highly structured languages such as Ada to more free-wheeling ones such as Lisp. Many are associated with specific settings or applications. Ada, for example, is often found in military projects, while Lisp is often associated with artificial intelligence. Some languages, such as HTML, the language used to describe the layout of World Wide Web pages, have a well-defined organizational structure, but have very little in the way of traditional program structure (there are no data types, for example).

In trying to understand a new language it is not only important to master its syntax, it is also vital to appreciate its style—the way in which that language can be used to accomplish specific goals. We have already reviewed the basic goals of JavaScript in chapter 1, "What Is JavaScript?" as well as contrasted it to the more structured Java language. This chapter describes the JavaScript language from both perspectives. A thorough description of its syntax is given, and some initial concepts on how to structure a JavaScript program are also introduced. Anyone who has programming in almost any modern declarative language,

such as C, C++ or Pascal, will feel immediately at home. In addition, HTML authors who have never programmed will be able to rapidly acquire JavaScript proficiency.

JavaScript Syntax

JavaScript is based on an action-oriented model of the World Wide Web. Elements of a Web page, such as a button or checkbox, may trigger actions or *events*. When one of these events occurs, a corresponding piece of JavaScript code, usually a JavaScript function, is executed. That function, in turn, is composed of various statements which perform calculations, examine or modify the contents of the Web page, or perform other tasks in order to respond in some way to that event. For example, pressing the `SUBMIT` button on an online order form might invoke a JavaScript function that validates the contents of that form to ensure that the user entered all the required information.

In this section, we examine the syntax of JavaScript from the bottom up. We begin with the most basic concepts of how to write a JavaScript statement, and what that statement does, and progress upward through more complex and powerful structures in subsequent sections, culminating in a detailed discussion of JavaScript functions and related concepts. [Chapter 3](#) "Events and JavaScript," explores in greater detail how these elements are tied into Web pages through events.

In general, the elements of a JavaScript program can be divided into five categories, as follows:

- Variables and their values
- Expressions, which manipulate those values
- Control structures, which modify how statements are performed
- Functions, which execute a block of statements
- Objects and arrays, which are ways of grouping related pieces of data together

This set of categories is very similar to many other languages. As we examine each of these elements in subsequent sections we will discover that JavaScript is somewhat minimalist in its approach. Many familiar elements, such as explicit data types (`int`, `String`, `REAL`), are missing or have been substantially simplified. However, JavaScript also provides a number of powerful object-oriented constructs which greatly simplify program organization. In this way, JavaScript has the expressive power of languages such as C or Java, while also having fewer rules to remember.

Variables and Values

One of the main differences between JavaScript and most other languages is that it does not have explicit data types. There is no way to specify that a particular variable represents an integer, a string, or a floating-point (real) number. Any JavaScript variable can be any of these—in fact, the same variable can be interpreted differently in different contexts.

All JavaScript variables are declared using the keyword `var`. A variable may be initialized, meaning that it is given a value when it is declared, or it may be uninitialized. In addition, multiple variables can be declared on the same line by separating their names with commas. For example, the statements

```
var x = 7
var y, z = "19"
var lk = "lucky"
```

declare a variable named `x` with initial value 7, an uninitialized variable `y` and variables named `z` and `lk` whose initial values are "19" and "lucky," respectively. It might seem that `x` is an integer, `z` and `lk` are strings, and `y` is some undefined quantity. In fact, the real story is a little more complicated than this. The value of each variable depends on the context in which it is used. This context is related to the order in which the variables are seen. As you might guess, the expressions

```
5 + x
lk + z
```

evaluate to 12 and "lucky19," seemingly confirming our suspicions about what they really are. However, it is also possible to form the expressions

```
lk + x
x + z
```

which evaluates to "lucky7" and 26, respectively. In the first expression, `x` has been interpreted as a string, while in the second, `z` has been interpreted as an integer.

TIP

JavaScript often attempts to treat all variables within a statement as if they had the same type as the first variable in the statement.

These examples illustrate two *critically* important points about the JavaScript language. First, while JavaScript does not have explicit data types, it does have implicit data types. Second, JavaScript has a set of conversion rules that allow it to decide how to treat a value based on the context in which it is used. The context is established by reading the expression from left to right. In the expression `x + z`, for example, `x` is implicitly a numerical value, so that JavaScript also attempts to view `z` as a number and perform the sum numerically. It succeeds, and the expected 26 results.

What would have happened if we had tried `x + lk`? The `x` variable occurs first on the left, and is really a number at heart. JavaScript thus tries to interpret the variable `lk` as a number, too. This is extremely unlucky, in fact, because "lucky" cannot be converted to a number (while `z`, the string "19," could). JavaScript reports an error if asked to evaluate `x + lk`. To understand JavaScript variables and values, therefore, it is necessary to understand its set of implicit types and how they may be converted to one another.

Before we enter into these details, let us consider one final example. In all the preceding cases, the uninitialized variable `y` was never used. What would be the value of an expression such as

$$x = z + y$$

Of course, as in all other programming languages, the result of using an uninitialized variable is never good. Since `y` has never been given a value, there is no way this expression can be evaluated. It may result in something seemingly innocent, such as `x` being assigned the value of `z`, as if `y` were zero. It may also result in something much more serious, such as the value of `x` becoming something strange, or, more likely, a JavaScript error occurring. This leads to the following common sense rule.

TIP

Initialize all JavaScript variables to meaningful default values. If a variable has no meaningful default, initialize it to `null`.

Implicit Data Types in JavaScript

There are five major implicit data types in JavaScript. A JavaScript value may be as follows:

- A number, such as `-5`, `0`, or `3.3333`
- A string, such as `"Click Here"` or `"JavaScript"`
- One of the logical values `true` or `false`
- A "non-atomic" JavaScript element, such as a function or object
- The special value `null`

Actually, it would be more correct to say that there are five categories of data type, since it is possible to distinguish two different types of numbers (integers and floating-point numbers), and many different types of JavaScript objects, functions, and other structured types. In fact, part II of this book, "JavaScript Objects," is entirely devoted to explaining the many different JavaScript objects.

Variables and Variable Names It is very important to distinguish between variables and their values. The statement `x = 10` contains two components: the variable `x` and the *literal* value `10`. A literal refers to anything that is referred to directly, by its actual value. A variable is just an abstraction that provides a way of giving names to values. Thus the statement `x = 10` says, "I am going to refer to the concrete (literal) quantity `10` by the abstract (variable) name `x`," just as you might say, "I am going to call this lumpy thing I'm sitting on a chair." This also leads to the following important piece of advice.

CAUTION

It is bad practice to change the implicit data type of a variable. If a variable is initialized to have a certain type (such as string) it should always have that type.

Thus, since we have started out with `x = 10` we should make sure that `x` always has some numeric value. There is no rule that prohibits us from later saying `x = "Fortran"`, but this will generally lead to confusion or programming errors in most cases. No one will stop you from calling that lumpy thing you are sitting on "bacon and eggs" but many of your guests may become confused if you do so.

One final rule about variable names: a valid JavaScript variable name must begin with a letter or with the underscore character (`_`). Case is important, so that `norl`, `NoRl`, `NORL`, and `_NORL` are all valid JavaScript variable names that refer to different variables.

Numerical Values There are two numeric types in JavaScript: integers and floating-point numbers. The rules for specifying both types are almost identical to those of C or C++ or Java. Integers may be specified in base 10 (decimal), base 8 (octal), or base 16 (hexadecimal) formats. The three forms are distinguished as follows, based on the first one or two characters:

- 1-9 followed by any set of digits is a decimal integer.
- 0 followed by any set of the digits 0-7 is an octal integer.
- 0x or 0X followed by any of 0-9, a-f, or A-F is a hexadecimal integer.

Any of the three forms can also start with a + or - sign. Thus, `-45` is a decimal integer, `017` is an octal integer, and `0x12EF5` is a hexadecimal integer. The minimum and maximum integers that can be used are implementation dependent, but at least 32 bits should be expected.

Floating-point numbers can be specified in either the standard decimal point (`.`) format or the engineering E-notation. Typical floating-point numbers should contain a decimal point or an exponent, which may begin with either `e` or `E`. A floating-point number may also have a + or - sign. `0.0`, `-1.4e12`, and `3.14159` are all valid floating-point numbers. The range of valid floats is again implementation dependent, but you should expect that any valid short floating-point number, as defined by the IEEE standard, is acceptable. (The IEEE is the Institute of Electrical and Electronics Engineers, a professional and standards-making organization.)

Note that the original LiveScript language attempted to treat all the numeric types the same. Since it has become JavaScript there has been a convergence toward the numerical types of the Java language, and the distinction between integer values, such as `5`, and floating-point (or real) values, such as `3.3333`, has increased.

CAUTION

LiveScript is now completely obsolete. It has been

replaced by its descendant, JavaScript. Avoid any code you encounter labeled LiveScript as it will almost certainly not work correctly.

NOTE

Watch out for changes in the way JavaScript handles numeric types. In the future, the distinction between integers, single precision floating-point types (floats), and double precision floating-point types (doubles) may become much sharper.

Strings In JavaScript, strings may be specified using either single quotes ('stuff') or double quotes ("otherstuff"). If you begin a string with one type of quote you must end it with that same form of quote—for example, "badstuff" is not a legal string in JavaScript. Strings may also be nested by alternating the types of quotes used. In fact, you must alternate single and double quotes if you wish to put one string inside another. Here is an example of several nested strings (with apologies to Rudyard Kipling):

"Oh, it's 'Tommy this' and 'Tommy that' and 'throw im out, the brute'"

As in C and Java, JavaScript strings may contain special combinations of characters, known as *escape sequences*, to denote certain special characters. The rules for this are still emerging, but it is probably safe to assume that all the escape sequences defined in C will be supported. Since you will almost always be using formatting directives of HTML (such as `
` for a line break) you will probably not use these directives very often. At the moment, the following sequences are supported:

<code>\t</code>	tab
<code>\r</code>	line feed
<code>\n</code>	return
<code>\f</code>	form feed (vertical tab)
<code>\b</code>	backspace

The special string `""` or `''` represents the zero length string. This is a perfectly valid string whose length is zero. This is the shortest JavaScript string; the length of the longest is, as usual, implementation dependent. It is reasonable to expect that most JavaScript environments will permit very long sonnets (or very short legislative measures) to be represented as single strings.

Logical Values The logical, or boolean, values `true` and `false` are typically used in expressions that test some condition to determine how to proceed. If that condition is met then one set of statements is executed; if it is not then another set is used instead. The first corresponds to the `true` condition, while the second represents the `false` condition. Not surprisingly, such expressions are known as *conditional expressions*. As you will see in the

"Operators" section there are several comparison operators, such as the equality test (`==`), which result in logical values.

It is possible to think of `true` as 1 and `false` as 0. In fact, JavaScript converts these logical values into 1 and 0, respectively. JavaScript also accepts any non-zero integer in place of `true`, for example, so that 5 and -3 can both be used as stand-ins for `true`. Many different programming languages follow this same convention. It should be avoided in JavaScript, as it can lead to type confusion.

The Value `null` The value `null` has a very special role in the JavaScript language. It is the value of last resort, so to speak, for every variable. For the beginning JavaScript programmer, its primary role will be in initializing variables that do not have any more meaningful initial value. For example, in the set of variable declarations given in the earlier "Variables and Values" section, to initialize `y` to some value, we should have actually written

```
var y = null
```

This prevents JavaScript errors that arise when an uninitialized variable is accidentally used in an expression that requires a value. It is important to realize that the value `null` does not give the variable `y` any implicit data type. `null` also has the property that it may be converted to a benign form of all the other types. When it is converted to a number it becomes 0, when it is converted to a string it becomes the empty string `""`, and when it is converted to a boolean value it becomes `false`. This is the one case where it is permissible to change the implicit data type of a variable after it is declared.

Therefore, statements such as

```
var lk2 = lk + y
var w = x + y
```

result in `lk2` having the value "lucky" (the same as `lk`) and `w` having the value 10 (the same as `x`). This is why the value `null` is an excellent way of initializing variables—it is guaranteed to be harmless.

Type Conversion

Several of the examples in the previous section use the `+` operator to combine different types of things. You may recall that when a string is combined with a number in the form

```
stringthing + numberthing
```

the number is converted to a string and the `+` operator then glues the two strings together (concatenation). However, if they are combined in the opposite order

```
numberthing + stringthing
```


then JavaScript attempts to convert the `stringthing` to a number and add it, numerically, to `numberthing`. If the `stringthing` can be converted to a string, such as `"-14"`, then all goes well; if it cannot then an error results. This illustrates the concept of implicit conversion in JavaScript.

We have already seen that some examples of implicit conversion are completely safe. `false` can be converted to 0, `"5"` can be converted to 5, and `null` can be converted to just about anything. However, some conversions are obviously invalid, and others might be questionable. Questions such as, "May the string `'3.0'` be legitimately converted to the integer 3?" are actually very difficult to answer with complete generality.

There are two approaches to handling this complex issue: use explicit conversion whenever possible, and use implicit conversion with great care. Both approaches should be used. A detailed study of explicit conversion is in chapter 5, "Built-In JavaScript Objects," beginning with the section "The String Object." For the moment we will use the following rules of thumb.

CAUTION

Use implicit conversion only when converting to a string form. Never use it to convert to numerical form. This is because attempts to convert a non-numerical quantity to a numeric form cause serious JavaScript errors, while conversion to string form generally does not.

You have probably already noticed that conversion to a string is always safe, at least for the data types we have encountered so far. In fact, this type of implicit conversion is a boon to the JavaScript programmer, since it avoids the tedious formatting directives that are necessary in many languages such as C. In JavaScript we can say

```
"This page has been accessed " + cnt + " times today"
```

without having to worry about the data type of the variable `cnt`. This construction will always give a valid string, and never an error.

The preceding Caution is also based on standard principles of defensive programming. There are many things that cannot be sensibly converted to a numerical form, so the prudent approach is to never try to implicitly convert anything to a number. There are several more robust approaches that can be used in case we have a string that we want to convert to numerical form. These are described in chapter 5. We will also see other exceptions to this rule as our mastery of JavaScript deepens.

Statements and Operators

The basic unit of work in JavaScript is the *statement*, as is the case in most programming languages. A JavaScript statement accomplishes work by causing

something to be evaluated. This can be the result of giving a value to a variable, by calling a function, by performing some sort of calculation, or any combination of these. We have already seen variable declaration statements, which not only create (declare) a new variable, but also give it an initial value, such as the following statement:

```
var x = 10
```

JavaScript programs, as mentioned at the beginning of this chapter, are collections of statements, typically organized into functions, which manipulate variables and the HTML environment in which the script itself works, in order to achieve some goal.

The Structure of JavaScript Statements

Before plunging into a detailed description of the various types of statements and the operators they use, let's examine one simple statement in excruciating detail. Consider the statement

```
y = x + 5
```

This statement contains three parts: the result *y*, the *operator* = and the *expression* *x + 5*. The result always occurs in the left side, since JavaScript always operates from left to right, and is often called the *lvalue*. The result must always be something that can be modified. It would be erroneous to write `null = x + 5`, for example, because `null` is a built-in, unchangeable component of JavaScript itself—it cannot be modified, so it can never appear as a result.

The operator = is the assignment operator, of course. It causes the expression on the right to be evaluated and its value given (assigned) to the result. The expression *x + 5* contains another operator, the + operator, which acts to combine *x* and 5 in some context-specific way. Since *x* is a number in this case, the + operator performs ordinary addition, and *y* gets the value 15. As we have already seen, if *x* had been a string, such as "bleh," then + would have acted as a string concatenation operator and *y* would be given the value "bleh5" instead. This is an example of *operator overloading*—the + operator can do different things in different situations. Many JavaScript operators are overloaded.

There is one final point to be made about this statement and about the structure of JavaScript programs in general. JavaScript has adopted a line-oriented approach to program flow. This means that it knows that a statement has ended when it reaches the end of a line. It is also possible to explicitly terminate a statement with a semicolon character (;). The statement `y = x + 5;` is identical in effect to the statement `y = x + 5`. This also means that you can, in fact, put multiple statements on a single line by separating each of them with a semicolon.

For those just starting out in JavaScript it is often a good idea to terminate each statement with a semicolon, and to also put only a single statement on each line. This might seem both redundant and extraneous, but it is well justified. The end of a line is often a purely

visual concept. Anyone who has ever used a word processor has undoubtedly encountered the situation where a very long line looks like two lines. Different platforms (Macintosh, PC, UNIX) also have their own unique ideas as to what the proper end-of-line characters are. It is much safer to put in the extra semicolon character and be explicit about the end of the statement than it is to rely on one's eyesight.

Operators

The set of operators that JavaScript uses is, once again, very similar to that of the C, C++, and Java languages. It provides a number of different ways of combining different values, both literals and variables, into expressions. Some operators require two elements to participate in the operation, and are referred to as *binary* operators. The `+` operator is a binary operator. Other operators require only a single participant (operand), and are known as *unary* operators. The `++` operator, which adds 1 to its operand, is a unary operator. Operators may also join forces to form aggregate operators, as we shall see next.

JavaScript operators may be classified into the following groups:

- Computational operators
- Logical operators
- Bitwise operators
- Assignment and aggregate operators

This grouping is purely functional, and is based on what the operators actually do. The next four subsections examine each type of operator in more detail. Table 2.1 summarizes the operators in each category and how they are used.

Table 2.1 A Summary of JavaScript Operations
Computational Operators

<code>+</code>	Addition, String Concatenation
<code>-</code>	Subtraction, Unary Negation
<code>*</code>	Multiplication
<code>/</code>	Division
<code>%</code>	Modulus
<code>++</code>	Preincrement, Postincrement
<code>--</code>	Predecrement, Postdecrement

Logical Operators

<code>==, !=</code>	Equality, Inequality
<code><, <=, >, >=</code>	Arithmetic and String Comparison
<code>!</code>	Logical NOT
<code>&&, </code>	Logical AND, Logical OR
<code>?</code>	Conditional Selection (ternary)

,	Logical Concatenation
Bitwise Operators	
&,	Bitwise AND, Bitwise OR
^	Bitwise eXclusive OR (XOR)
~	Bitwise NOT
<<,>>,>>>	Shift Left, Shift Right, Unsigned Shift Right
Assignment Operators	
=	Assignment
OP=	Aggregate Assignment (+,-,*,/,%,&, ,^,~,<<,>>,>>>)

Computational Operators The computational operators are addition (+), subtraction (-), division (/), multiplication (*), modulus (%), increment (++), and decrement (--). These operators are often used in performing arithmetic computations, but do not forget that the + operator is overloaded; it also has the extremely important role of string concatenation.

The first five computational operators have their standard mathematical meanings. They add, subtract, divide, or multiply two numeric quantities. By combining two quantities using one of these operators the result is made as precise as possible. If an integer is added to a floating-point number, the result is a floating-point number. The following four statements illustrate the use of these operators:

```
x = 4 + y;
y = 5.5 - z;
z = 10 / w;
w = 1.4e5 * v;
```

Note that division of integer quantities result in an integer result, so that if *w* had the value 4 in the third statement, *z* would get the value 2, not 2.5. Note also that the - operator may also be used as a unary operator to compute the negative of a numeric quantity:

```
n = -m;
```

This has exactly the same effect as if we had multiplied *m* by -1.

The modulus operator (%) is used to compute the remainder from a division. Although it can be used with floating-point numbers, it is typically used with integers, so that `21 % 4` evaluates to 1. The modulus operator always gives a remainder that has the same sign as the corresponding quotient, so that `-21 % 4` evaluates to -1, not 3.

The increment and decrement operators are conveniences created to simplify the very common operations of adding or subtracting one from a number. Both these operators are

unary and come in two forms: prefix and postfix. The expression `++x` is the preincrement form of the `++` operator, while `x++` is the postincrement form. This leads to a subtle and often misunderstood point about the increment and decrement operators.

Supposing that `x` has its usual value 10, consider the two statements

```
y = ++x;
z = x++;
```

These look very similar, but are in fact very different. After both of these statements have been executed, `x` has the value 11. However, `y` ends up with the value 11 while `z` has the value 10. Why? The reason has to do with the complex issue of what order the operators `++` and `=` are evaluated in these two statements. In the first statement, the `++` is evaluated first, so that `x` attains the value 11, and then the assignment `=` is evaluated, so that this value is passed on to `y`. In the second statement, the assignment operator `=` is applied first, so that `z` becomes 10, the current value of `x`, and then the `++` is applied to `x`, so that it advances to 11. The same rule applies to the decrement operator (`--`).

This might seem like it is a violation of the rule of left-to-right evaluation, and it is. Even though the equal sign is to the left of the preincrement operator (`++`) in the first statement, the `++` operator takes effect first. This is an example of *operator precedence*, the order in which multiple operators are applied. This complex topic is discussed in more detail in the "Order of Evaluation" section later in the chapter.

Logical Operators Logical operators in JavaScript are used either to carry out some form of test, or to combine the results of more than one such test. They are often referred to as conditional operators. The logical operators that perform a test of some sort are the equality/inequality operator (`==` and `!=`), the comparison operators (`<`, `<=`, `>`, and `>=`), and the logical negation operator (`!`). The operators that combine logical values are logical AND (`&&`) and logical OR (`||`). Finally, the conditional operator (`?`) and the comma operator (`,`) are also combining operators, although they are only vaguely logical operators.

Equality Operators The binary equality (`==`) and inequality (`!=`) operators are used to test if two quantities are the same or different. These operators are overloaded. On integers, they test for strict equality or inequality. On floating-point numbers, they test to see if the two quantities are equal within the precision of the underlying floating-point type. On strings, they test for exact equality—recall that case is significant in JavaScript strings. These operators all return a boolean value, either `true` or `false`.

For example, if `x` has the value 10, `y` has the value 3.0, and `z` has the value "barney," then `x == 10` is true, `y != -5.0` is also true, and `z == "fred"` is false. Unfortunately, even operators as simple as these can be a source of error. It is regrettable that the logical operator `==` looks so much like the assignment operator `=`. Consider the following incorrect code fragment:

```
if ( x = 3 ) {
    stuff...
```

The purpose of this code is almost certainly to test the value of the variable `x` against the constant `3`, and execute the `stuff` if that test succeeded. This code fails to realize that purpose in two very dramatic ways, just by inappropriately using `=` instead of `==`.

First of all, `x = 3` always gives `x` the value `3`, no matter what its previous value was. Instead of testing `x` using `==`, we have altered it with `=`. Second, the value of the expression `x = 3` is the value of its left side, namely `3`. Even though `3` is not a true logical value, it is treated as true by the `if` statement (`if` is described in greater detail in the section, "Control Structure," later in this chapter). This means that `stuff` will always be executed, rather than only being executed when `x` has the prior value `3`.

This type of error occurs in every programming language in which similar operators are used for very different purposes. In this case, we could have adopted another rule of defensive programming and said

```
if ( 3 = x ) {
  stuff...
```

In this case, our typing mistake (`=` instead of `==`) leads to an error, rather than resulting in a subtle programming flaw. Since `3` is a constant, it can never appear on the left side of an assignment, but it is quite capable of appearing on the left side of a logical test. Said another way, since `x == 3` and `3 == x` are completely equivalent, the form `3 == x` is preferable. If it is mistyped as an assignment statement (`3 = x`) it leads to an immediate error rather than one which might take hours of debugging to uncover. This leads to the following advice.

TIP

When testing for equality always put constants on the left, especially `null`.

There is another subtle evil about the (in)equality operators when they are used with floating-point numbers. It is very tricky to make floating-point arithmetic completely independent of the underlying machine. This means that `z == 3.0` might be true on one machine but false on another. It can also lead to seemingly absurd results such as `3.0 != 3.00` being false while `3.0 == 2.9999999` is true. A remedy for this problem is presented at the end of this section.

Comparison Operators The comparison operators (`<`, `<=`, `>` and `>=`) also operate on both numbers and strings. When they act on numbers they perform the usual arithmetic comparisons, yielding boolean values, as with the equality operators. When they act on strings they perform comparisons based on dictionary order, also known as lexicographic order. If a string `str1` occurs earlier in the dictionary than a second string `str2` then the comparison `str1 < str2` (and also `str1 <= str2`) will be true. For example, `"barney" < "fred"` is true, while `"Franklin" < "Delano"` is false.

The Negation Operator The logical negation operator (!) is used to reverse the sense of a logical test. It converts true to false and false to true. If `x < 15` is true then `!(x < 15)` is false, and vice versa. Note that ! may also be used with integer values, so that `!0` is true, while `!5` is false. As in other cases, this use of the ! operator violates type boundaries, and should be avoided.

Boolean Logical Operators The logical AND (&&) and OR (||) operators are among the most powerful operators in JavaScript. Both may be used to combine two or more conditions into a composite test. The logical AND of a set of conditions is true only if all of its component conditions are true. The logical OR of a set of conditions is true if any of its component conditions are true. Thus

```
( x < 17 ) && buttonPressed && ( z == "Meta" )
```

is true precisely when `x` is less than 17 *and* the boolean variable `buttonPressed` is true *and* `z` is exactly equal to the string "Meta." Similarly,

```
( x < 17 ) || buttonPressed || ( z == "Meta" )
```

is true if one or more of the three conditions is true.

Lazy Evaluation JavaScript uses a lazy variant of its usual left-to-right evaluation rule with the && and || operators. This lazy evaluation (or short circuit evaluation) rule states that JavaScript stops trying to evaluate the expression as soon as its value is known.

To see how this works, suppose that `x` has the value 20, `buttonPressed` is true, and `z` is the string "Hyper". Since `(x < 17)` is false the second and third conditions in the logical AND statement are never evaluated. This is because false && anything is always false, so the value of the first expression must be false. Similarly, the second statement stops as soon as `buttonPressed` is evaluated. Since true || anything is always true, the second expression must be true.

Lazy evaluation can be both a boon and a curse. Suppose that `"digofpi(1000000000)"` is a function that computes the billionth digit of *pi*.

The expression

```
( x < 25 ) || ( digofpi(1000000000) == 3 )
```

does not actually try to compute the billionth digit of *pi* if `x` is 20, because the expression is already known to be true, and `digofpi()` is never called. As an additional example, consider the following expression:

```
( x < 25 ) && beaupage()
```

Suppose that `beaupage()` is a function that displays a beautiful Web page. If `x` is 30 this page will never be seen, because the first part of expression `(x < 25)` is already known to be false. As a result the function `beaupage()` is never called. We revisit this phenomenon in the "Functions and Objects" section at the end of this chapter. For the moment, it is wise to be aware of lazy evaluation.

Fuzzy Comparison The logical AND and OR operators also provide us with one solution to the problem of floating-point comparison. While it may not be possible to ever determine if `x` is exactly equal to the constant 3.0, you can be certain that it is close using a combined test such as

```
( x - 3.0 ) < epsilon || ( 3.0 - x ) < epsilon
```

where `epsilon` is some suitably small value, such as 0.001. This form of test is often referred to as a *fuzzy comparison*.

CAUTION

Floating-point arithmetic is not an exact science. Avoid exact comparison tests such as `==` and `!=`; use fuzzy comparisons instead.

The Comma and Question Mark Operators The final two operators in the logical category are the conditional operator (`?`), often called the question mark operator, and the comma operator (`,`). These two operators are only vaguely logical, but they don't readily fall into any of the other categories either.

The conditional operator is the only trinary (3 operand) operator in JavaScript. It is used to select one of two possible alternatives based on a conditional test. The syntax for this operator is

```
( conditionthing ? truealt : falsealt )
```

If the `conditionthing` is true then the value of this expression is `truealt`; otherwise it is `falsealt`. Note that the colon (`:`) separating the true alternative from the false alternative is mandatory. This can be used to select an appropriate alternative and simplify code, as in this example:

```
printme = ( errorCode == 0 ? "OK" : "error" );
```

This expression makes the variable `printme` have the string value "OK" in case the variable `errorCode` is 0; otherwise, it is set to "error". The question mark operator is often a fast way to select one of two choices when a control structure would be unnecessarily cumbersome.

Finally, the lowly comma operator can be used to force the evaluation of a set of expressions. All intermediate results are discarded, and the value of the very last expression on the right is returned. For example, the expression

```
b = (d = digofpi(1000000000)), beaupage(), (x < 17);
```

always computes the billionth digit of *pi* and assigns it to the variable *d*, always displays the beautiful page, always compares *x* against 17, and only returns the result of that comparison since *x < 17* is the rightmost expression. The result of that comparison is assigned to the boolean variable *b*. This might seem like a clever way to outwit JavaScript's lazy evaluation, but it would be clearer to simply write

```
d = digofpi(1000000000);
beaupage();
b = ( x < 17 );
```

In general, the comma operator is only useful when it is inside a `for` loop (see "Control Structures," later in this chapter), and should otherwise be ignored.

Bitwise Operators In many situations you do not need to know, nor do you wish to know, the precise binary representation of values in your program. There are some situations, however, in which it is absolutely essential to operate at the lowest possible level and deal with the individual bits of a particular value. This often arises in mathematical applications, for example, or when precisely manipulating color values. The bitwise operators are used for this purpose. Table 2.2 shows JavaScript's bitwise operators. Note that all are binary, except for bitwise NOT, which is unary. Each operates on its operands one bit at a time.

Table 2.2 JavaScript's Bitwise Operators

Operator Name	Symbol
Bitwise AND	&
Bitwise OR	
Bitwise XOR	^
Bitwise Left Shift	<<
Bitwise Signed Right Shift	>>
Bitwise Unsigned Right Shift	>>>
Bitwise NOT	~

Bitwise AND (&) examines each bit position in each of its operands. If both operands have a 1 bit in a given position, then that bit will also be set to 1 in the result. In all other cases, the output bit position is zero. For example, suppose *x* = 0x00001234 and *y* = 0x8000ABCD. Then *z* = *x* & *y* will have the value 0x00000204. You can see this more easily by writing *x* and *y* in base 2 (binary) notation, and looking for those positions in which both *x* and *y* are 1, as shown in the first part of figure 2.1.

Figure 2.1 : JavaScript's bitwise operators operate on each bit separately.

Note that x and y only have the same bits set in highlighted positions, so that those are the only bits set in their logical AND z . In this way, bitwise AND is the bit level analog of the logical AND. Bitwise OR ($|$) is similar. If either bit is 1 in any bit position, then that bit will be 1 in the result. Thus the value of $w = x | y$ will be `0x8000BBFD`, as you see in the middle part of figure 2.1.

Each bit is set in w if either or both of the corresponding bits in x and y is set. The bitwise XOR (exclusive OR) (\wedge) operator is a variation on the bitwise OR operator. It sets a bit in the result if either bit in the operand is set, but not both. The value of $v = x \wedge y$ is `0x8000B9F9`, as shown at the bottom of figure 2.1.

These three operators may also take more than two operands, so that it is possible to write a very long expression such as

```
n = ( a & b & c & d & e );
```

which operates from left to right, as usual. This expression takes the bitwise AND of a and b , ANDs that result with c , ANDs that result with d , and finally ANDs that result with e . The final result is saved in the variable n .

CAUTION

The bitwise AND ($\&$) and OR ($|$) operators bear a shocking similarity to their logical counterparts $\&\&$ and $||$. This can lead to painfully undetectable errors. The same care that is exercised with $=$ and $==$ should also be used with these operators.

The unary bitwise NOT operator (\sim) changes each 0 bit in its operand to a 1 bit, and each 1 bit in its operand to a 0 bit. The bitwise NOT of x will have the value `0xFFFFEDCB`:

x	0000 0000 0000 0000 0001 0010 0011 0100
$\sim x$	1111 1111 1111 1111 1110 1101 1100 1011

While $\&$, $|$, \wedge and \sim operate on bits in place, the shift operators \ll , \gg , and \ggg are used to move bits around. The left shift operator (\ll) shifts a set of bits to the left by a specified number of positions, while both \gg and \ggg moves that set of bits to the right in two potentially different ways. For example, let us evaluate these three expressions:

```
xleft = x << 5;
ysright = y >> 3;
yusright = y >>> 3;
```

The first of these shifts each bit in x to the left five positions. Zero bits are tacked on at the right, while the bits that are shifted out at the left are lost when they exceed the overall 32-bit length. So the value of $xleft$ must be $0x00024680$. The signed right shift operator acts in almost the same way. Each bit of y is shifted to the right three positions. Bits on the right edge of y are lost as they are shifted out. However, rather than shifting in zeros at the left side of y , the most significant bit of y , which happens to be 1 in this case, is shifted in. The resulting value of $ysright$ is $0xF0001579$.

This might seem counterintuitive, but it makes good mathematical sense, since it preserves the sign of the operand. If y is negative (most significant bit set, as in our example) then any signed right shifted version of y will also be negative. Similarly, if y had been positive (most significant bit equal to 0) then any right shifted version of y would have been positive. The unsigned right shift operator ($>>>$) does not preserve the sign of its operand; it always shifts 0 bits in at the left edge. The value of $yusright$ is therefore $0x10001579$. The shift processes used to compute $xleft$, $ysright$, and $yusright$ are shown in figure 2.2.

Figure 2.2 : *JavaScript's shift operators move bits to the right or left, and are equivalent to multiplication or division by a power of two.*

Since all the bitwise operators act at the bit level, chaos can result if they are applied to a variable that is not an integer. Floating-point numbers are particularly sensitive, since an arbitrary bit pattern need not correspond to a valid floating-point number.

CAUTION

Never perform bitwise operations on floating-point numbers. Your code will be unportable, and floating-point exceptions may result.

Assignment and Aggregate Operators Our tour of JavaScript operators concludes with the assignment operator and its aggregates. You have already seen many examples of that most fundamental of all operators, the assignment operator ($=$). You are well aware that it is used to assign the result of an expression or value on the right side of the $=$ sign to the variable or lvalue on the left side of the $=$ sign.

In JavaScript, as in C, C++, and Java, you can also combine the assignment operator with any of the binary computational and logical operators. The expression

```
Left OP= Right ;
```

is just a shorthand for the expression

```
Left = Left OP Right ;
```


where *OP* is any of the operators `+`, `-`, `/`, `*`, `%`, `&`, `|`, `^`, `<<`, `>>`, or `>>>`. So, to add 7 to *x*, multiply *y* by 19.5, OR *z* with 0xAA7700, and perform an unsigned right shift of 10 bits on *w* you can write

```
x += 7;
y *= 19.5;
z |= 0xAA7700;
w >>= 10;
```

These compact expressions replace the wordier versions `x = x + 7`; `y = y * 19.5`, and so forth.

Order of Evaluation In elementary school math you were probably confronted with questions such as, "What is the value of `3 + 4 * 5`? Is it 23 or is it 35?" This was your first exposure to the concept of order of evaluation, or operator precedence. You probably remember that multiplication has a higher precedence than addition, so that the correct answer is 23. The same issue arises in almost every programming language with the concept of operators—which comes first?

There are two approaches to this issue. The first involves learning, or attempting to learn, the operator precedence table. The more operators there are, the more rules there must be in this table. The second approach is to simply ignore the issue completely and explicitly group your expressions using parentheses. Never write `3 + 4 * 5`. Always write `3 + (4 * 5)` or even `(3 + 4) * 5` if that is what you want.

This recommendation is very much like several others in this chapter. It trades the effort (and perhaps some readability) of using the explicit parenthesized form against the promise that the order of evaluation will always be exactly as you wrote it. Incorrect order of evaluation is almost certainly the second most common source of programming error in JavaScript (confusing `=` and `==` is the first). For the daring, figure 2.3 shows the operator precedence table for JavaScript. For everyone else, the following rule of thumb is recommended.

Figure 2.3 : *Use the operator precedence table to determine the order of evaluation.*

TIP

Use parentheses to explicitly specify the order of evaluation in expressions containing more than one operator

There is one case in which no amount of parentheses will help. When using the increment (`++`) and decrement (`--`) unary operators you must simply know that preincrements and predecrements always happen before anything else.

Comments in JavaScript Code

All professional code should have comments that clearly indicate the purpose and logic behind each major section of the code. JavaScript offers two comment styles—the original comment style from C and the single line comment style from C++ and Java.

C style comments are typically used to document major functions or code blocks. Because a C comment may extend over multiple lines it is ideal for detailed discussions of important parts of the code. A C comment begins with `/*` and ends with `*/`. Our aesthetically pleasing function `beaupage()` might begin with a thorough description of just what makes it so beautiful, as follows:

```
/*
   The function beaupage() draws a stunningly
   beautiful
   Web page by performing the following nineteen
   steps.
   ... list of the 19 steps
*/
```

By contrast, C++ style comments are most suitable for short, pithy descriptions which will fit on a single line. A C++ style comment begins with `//` and ends at the end of the current line. Critical variables, for example, might merit a Java style comment indicating how they will be used, as follows:

```
var done = false; // set to true when we are all done
```

Both comment styles may be mixed freely in the same JavaScript program. However, such comments should never be nested, as this can lead to confusion. Also, the temptation to use HTML style comments (`<!--` and `-->`) should be strongly resisted, for reasons which will become clear in the next chapter.

TROUBLESHOOTING

I have just written my first JavaScript program. Everything looks fine, but the code does nothing. What is wrong? Here is the code:

```
/* My first JavaScript program *?
...many lines of code not shown
/* End of my first JavaScript program */
```

Your comments are well thought out and informative. Unfortunately, your very first comment begins with `/*` but does not end with `*/`. You have inadvertently typed `*?` instead of `*/`, so that the comment does not end until very far down in your program. When you use C style comments always make sure that they match.

Control Structures

At this point, you have had just enough of the JavaScript language to declare variables, perform assignments, and do various types of arithmetic, string, and logical calculations. You are not yet able to write any meaningful code because you do not have any higher level constructions. In this section, we will consider various methods of controlling the way in which statements are executed. The next section will expose the highest level of JavaScript—its functions and objects.

There are three types of control structure in JavaScript, as follows:

- `if`
- `while`
- `for`

These three control structures are very similar. Each is introduced by a keyword (`while`, `for`, and `if`, respectively) and each manipulates a *block* of JavaScript statements. A block is introduced by a left brace (`{`) and terminated with a right brace (`}`). There can be as many JavaScript statements between `{` and `}` as you wish, or as few. A block of code can even be empty, with nothing between the braces. In many ways, a block of statements is like a single gigantic statement. In particular, block structured constructs are often all or nothing—either the entire contents of the block are executed, or none of it is. Since blocks behave like single statements, it is also possible to put blocks inside other blocks, in a nested fashion.

As you will see, each of the three control structures has its own specific format and its own special uses, although it is often possible to achieve the same results using any of the three types, with varying degrees of elegance.

The `if` Statement

The `if` statement is used to conditionally execute a single block of code. It has two forms, the simple `if` statement and the `if...else` statement. The simple `if` statement consists of a conditional expression, known as the `if` test, and a block of code which is executed if that expression evaluates to a boolean `true`. An example of an `if` statement follows:

```
if ( condstmt ) {
    zero or more statements
}
```

The block of code within the braces is often called the `if` block. The conditional statement `condstmt` can be any expression that yields a logical value. Note that numerical expressions may also be used; 0 is construed as `false` and all other values are taken to be `true`. As stated earlier, an `if` statement should be considered a single statement. Code blocks are not traditionally terminated with a semicolon, although there is no harm in doing so. Listing 2.1 shows an example of a simple `if` statement.

Listing 2.1 The *if* Control Structure

```

if ( ( x < 10 ) && ( -10 < x ) ) {           //
if test
    y = ( x * x * x );                       //
1: cube of x
    ystr = "The cube of " + x + " is " + y; //
2: informative string
}

```

In this example, the value of *x* is tested to see if it is less than 10 and also greater than -10. If the result of this test is *true* then the variable *y* is set equal to the expression *x * x * x*, known mathematically as the cube of *x*, in statement 1 (labeled 1:). The variable *ystr* is then set to a string that expresses this cubic relationship between *x* and *y*, in statement 2. If *x* fails either of the two tests in the *if* test then neither of the two statements in the *if* block are executed.

It is easy to see even in this simple example that it is often desirable to have a contingency plan in case the *if* test is *false*. This leads to the second form of the *if* statement, the *if...else* control structure. In this form, one block of code is executed if the *if* test passes, and a second block is executed if it fails. The format of this type of *if* statement is as follows:

```

if ( condstmt ) {
    ifblock of statements
} else {
    elseblock of statements
}

```

NOTE

In the current version of JavaScript, the placement of the braces is important. The opening brace ({) should be on the same line as the *if* keyword. If an *else* clause is present, the closing brace () of the *if* and the opening brace ({) of the *else* should both be on the same line as the *else* keyword. Other placements of the braces are allowed, but may not be understood.

In this form of the *if* statement, the *if* block is still executed if *condstmt* is *true*. However, in this case, the block of code following the *else* is executed if *condstmt* is *false*. Listing 2.2 shows an enhanced version of the code from listing 2.1 using the *if...else* form.

Listing 2.2 The *if...else* Control Structure

```

if ( ( x < 10 ) && ( -10 < x ) ) {
// if test
    y = ( x * x * x );           //
1: cube of x
    ystr = "The cube of " + x + " is " + y; //
2: informative string
} else {
// false case
    y = null;                   //
3: be paranoid; give

y a value
    ystr = "Cannot compute the cube of " + x; //
4: explain the failure
}

```

In this example, statements 1 and 2 are still executed if *x* meets both tests in the *if* test. If either test fails then statements 3 and 4 in the *else* block are executed instead. Statement 3 is another example of defensive programming. The variable *y* is given a value, albeit a meaningless value. This is done so that if *y* is used later it will be guaranteed to have some value (even if we forgot to initialize) regardless of whether the code flowed through the true part of the *if* (the *if* block) or the false part of the *if* (the *else* block).

Observe that *ystr* also gets a value no matter which of the two blocks is used. In the *true* case it has the informative string documenting the cube of *x*; in the *false* case it has a string indicating that the cube of *x* could not be computed. Since *ystr* will presumably be displayed to the user at some point, it is worthwhile to provide an error message. This is an example of *parallel code design*. Each conditional path modifies the same set of variables. For a simple case, such as listing 2.2, it is easy to ensure that this happens. There are only two variables, *y* and *ystr*, and we can see exactly where they are set in every case. For more complicated, nested conditional expressions, it can become almost impossible to observe every variable in every case. Parallel code design is a good goal to strive for nonetheless.

The while Statement

The *while* statement is used to execute a block of code while a certain condition is true. The format of the *while* statement is as follows:

```

while ( condstmt ) {
    zero or more statements
}

```

The condition clause *condstmt* is evaluated as a logical expression. If it is *true* then the block of statements between the braces is executed. The flow of control then loops back to the top of the *while* statement, and *condstmt* is evaluated again. This process continues until the *condstmt* becomes *false*, or until some statement within the block forces it to

terminate. Each pass through the block of code is called an *iteration*. Figure 2.4 illustrates the basic structure of a `while` statement.

Figure 2.4 : *JavaScript's while control structure executes a block of statements conditionally.*

The first fundamental difference between a `while` statement and an `if` statement is that the `while` block may be executed many times, while the `if` or `else` blocks are executed once at most. You might well wonder how a `while` statement ever terminates. The code shown in listing 2.3 illustrates a simple situation in which the `while` block eventually leads to the `condstmt` becoming false.

Listing 2.3 A `while` Loop That Adds a Sequence of Numbers]

```
var x = 1;
var xsum = 0;

while ( x <= 10 ) {                // loop until x
    is greater than 10
        xsum += x;                // add x to the
    running sum xsum              // increment x
    x++;
}
```

This code accumulates the sum of all the integers between 1 and 10, inclusive, in a variable called `xsum`. `x` starts out as 1, so that `xsum` initially becomes 1 as well. `x` is then incremented to 2 by the `x++` statement. That value is then added to `xsum`, so that it becomes $1 + 2 = 3$. This process continues until `x` finally becomes 11 and the `x <= 10` condition is false. `xsum` at this point has the value $1 + 2 + \dots + 9 + 10 = 55$. Thus, the loop terminates. Note that it is critically important to initialize `xsum` to 0. If `xsum` is not initialized at all then the statement `xsum += x`, which is just shorthand for `xsum = xsum + x`, gives an error. If `xsum` is initialized to something other than 0, the final result contains that initial value, and is not just the sum of the integers from 1 through 10.

Listing 2.3 shows one way in which a `while` loop can terminate. Statements within the block may cause the conditional statement to become false. It could also happen that the conditional statement at the top of the `while` was never true, so that the statements within the block are not executed even once. If `x` had started with the value 20 in this example then the `while` test would have been immediately false, and the statements `xsum += x` and `x++` would have never been executed. In this event, `xsum` would retain its initial value of 0.

Using the `break` Statement There is a third way for a `while` loop to terminate. If the special statement `break` is encountered inside the `while` block, the loop is forced to terminate immediately. No further statements are executed and the `condstmt` is not

retested. Execution continues with the first statement after the end of the `while` block. Listing 2.4 gives an example of the use of the `break` statement.

Listing 2.4 A `while` Loop with an Internal `break` Statement

```
var x = 1;
var xoddsum = 0;
var xtmp = 0;
var lastx = 0;

while ( true ) {           // 1: loop forever
    (well, almost)
    xtmp = xoddsum + x;     // 2: compute a
    trial sum
    if ( xtmp > 100 )       // 3: if it is too
    large, then...
        break;             // 4: we are done
    xoddsum += x;           // 5: add x to the
    running sum xoddsum
    x += 2;                 // 6: increment x by 2
}
lastx = x;                  // 7: save the final
value of x in the variable lastx
```

The test clause of this `while` (statement 1) is `true`, which, you might well suspect, is always `true`. This means that there is no way for this loop to terminate unless it is forced to do so by a `break` statement. In statement 2 a temporary sum is formed in the variable `xtmp`. This sum is tested against the limit 100 in statement 3; if `xtmp` exceeds it then statement 4, the `break` statement, is executed, and the loop terminates. If the test fails (`xtmp` is still less than 100) then the real sum is formed in statement 5. (Note that it would have been equivalent, and slightly more efficient, if we had written statement 5 as `xoddsum = xtmp`.) In statement 6, `x` is incremented by 2.

What does this `while` loop do? It keeps adding up numbers, odd numbers in fact, until the sum is less than 100. When the next sum would have exceeded 100, the `if` test succeeds, the `break` is executed, and the flow of control of the program reaches the first statement after the entire `while` block, namely statement 7. This statement saves the last value of `x` in a different variable, `lastx`. So this construction computes the largest sequence of odd numbers that can be added without having the sum exceed 100. You can easily determine for yourself that the value of `lastx` must be 21, since $1 + 3 + \dots + 21 = 100$ exactly, while $1 + 3 + \dots + 21 + 23 = 123 > 100$.

The Perils of Infinite Loops Listing 2.4 not only illustrates the use of `break`, it also shows two other elements worth noting. First, listing 2.4 contains a nested conditional: there is an `if` statement inside the `while` block. This sort of construct is extremely common, and many

levels of nesting are not at all unusual. Second, this example has another very common but somewhat troublesome feature. Since the `while` test is always `true`, there is no way for the `while` to terminate unless the `break` statement is executed. In the preceding example, it was terminated quite quickly. Suppose, however, that statement 6 had been incorrectly entered as `x -= 2`. In this case, `xoddsum` would be getting constantly smaller and `xtmp` would never exceed 100. This type of error is known as an *infinite loop*. Listing 2.3 is not immune either, even though it has a conditional test rather than a blanket `true`. If the final statement of that example had been mistyped as `x-`, it would never terminate either.

Naturally, infinite loops must be vigorously avoided. They will only terminate when some kind of internal error happens (such as an arithmetic overflow when something becomes too large) or as a result of user intervention. Unfortunately, there is no foolproof way to write a `while` statement (or a `for` statement, as we shall see shortly) that is guaranteed to be correct. JavaScript is no different than any other programming language in this respect. However, the following general principles will reduce the opportunity for error:

- Avoid `while (true)` whenever possible
- Have at least one way of exiting the loop body

If the `while (true)` construction is used, then the logic that exercises the `break` statement must be correct. If this logic isn't correct then the loop will never terminate. If you restrict your use of `while (true)` you will have fewer infinite loops. The second suggestion is based on the observation that the more chances there are to exit the loop, the less likely it is that the loop will last forever. Listing 2.5 shows a modified version of listing 2.4 in which we have moved the test on the sum to the `while` clause itself, and have also added a very paranoid test on the number of times through the loop (the variable `loopcount`).

Listing 2.5 An Improved Form of Listing 2.4

```
var x = 1;
var xoddsum = 0;
var lastx = 0;
var loopcount = 0;

while ( ( xoddsum + x ) < 100 ) {           // 1:
loop while sum is < 100
    xoddsum += x;                          // 2: add x
to the sum xoddsum
    x += 2;                                // 3:
increment x by 2
    if ( ++loopcount > 1000 )              // 4: if
we're working too late..
        break;                            // 5: quit
}
```

```

lastx = x;                                // 6: save the
final value of x in                        lastx

```

This version satisfies both rules. Of course, the test in statement 4 is completely unnecessary. The code is simple enough that we can reassure ourselves that it is correct and will not go into an infinite loop. Once you are writing slightly more complicated while loops, you will find that there are usually multiple possible error conditions that arise. Every time you test for an error you should consider using a `break` statement.

You will often see `while (true)` written as `while (1)`. These are equivalent, since `true` has the numerical value 1, but the latter form is sloppy. The conditional portion of a `while`, `if`, or `for` statement should always be a true logical expression.

Using the `continue` Statement There is another special statement that may be used inside while loops: the `continue` statement. The `continue` statement is used to force the flow of control back to the top of the while loop. When a `continue` statement is seen, all statements between it and the end of the while block are skipped, and execution continues at the top of the while. Listing 2.6 shows a simple use for the `continue` statement.

Listing 2.6 A `continue` Statement Returns to the Top of a `while`

```

var x = 0;
var xsum = 0;
var loopcount = 0;

while ( loopcount++ < 100 ) {           // 1: loop
100 times
    x++;                                // 2: increment x
    if ( ( x % 5 ) == 0 )                // 3: if x is
divisible by 5
        continue;                       // 4: skip it
    xsum += x;                           // 5: otherwise,
add x to xsum
}

```

This example adds up every number between 1 and 100 that is not divisible by 5. The numbers that are divisible by 5 are skipped by virtue of statements 3 and 4. Statement 3 computes the remainder when `x` is divided by 5. If that remainder is 0 then `x` must be evenly divisible by 5. In that case, the conditional in statement 3 is true, and statement 4 is executed. The `continue` statement causes execution to continue back to the top of the loop at statement 1. This means that statement 5 is not executed, so the sum always misses those values of `x` which are divisible by 5, and only those values.

Many programmers would write line 3 as `if (! (x%5))`. While this style is very common, it is also confusing and a potential source of error. One problem with this form is that it confuses JavaScript types by using the numerical value `x%5` as if it were a logical value. This form also hides the explicit test for zero of listing 2.6. While this `!` form is more compact, it is also more error prone, and should be avoided.

One striking difference between this listing and previous ones is that `x` is initialized to 0, not 1, and `x` is incremented at the top of the loop, not at the bottom. If the `x++` were at the bottom, what would happen? The values 1, 2, 3, and 4 would all be gleefully added into `xsum`. When `x` reached 5, however, statement 3 would be true, the `continue` in statement 4 would be executed, and both `xsum += x` and `x++` would be skipped. `x` would stay equal to 5 forever! Since the `x++` statement is critical to the correct functioning of the loop, it must occur before the `continue`. If it occurs after the `continue` it will be skipped.

CAUTION

Any statement that must be executed on every pass through a loop must be placed before any `continue` statements.

The for Statement

The `for` statement is the most powerful and complex of the three flow control constructions in JavaScript. The primary purpose of the `for` statement is to iterate over a block of statements for some particular range of values. The `for` statement has the following format:

```
for ( initstmt; condstmt; updstmt ) {
    forblock
}
```

The `for` clause, as shown, has three parts, separated by two mandatory semicolons. The `initstmt` is typically used to initialize a variable, although any valid statement may be used in this position. The `initstmt` is always executed exactly once, when the `for` statement is first encountered. The `condstmt` is a conditional test, and serves exactly the same function as in the `while` statement. It is tested at the top of each loop. The `for` statement terminates when this condition evaluates to `false`. The `updstmt` is executed at the bottom of each loop, as if it were placed immediately after the last statement in the `for` block. It is typically used to update the variable that is initialized by the `initstmt`.

Listing 2.7 shows a simple example of a `for` statement. In fact, the code in this listing accomplishes exactly the same task as the code in listing 2.3. Note that this code does not bother to initialize `x` when it is declared. This is because the `initstmt` part of the `for` loop sets it equal to 1 immediately.

Listing 2.7 Adding Up a Sequence of Numbers Using *for*

```

var xsum = 0;
var x;

for ( x = 1; x <= 10; x++ ) {           // 1: loop
while x is <= 10
    xsum += x;                         // 2: add x to xsum
}
```

In many ways, the `for` statement is very much like a fancy version of the `while` statement. Many of the observations that were made for `while` also hold true for the `for` statement. In particular, it is possible to use the `break` and `continue` statements within a `for` loop. One of the advantages of a `for` loop is that its update statement is executed on every pass through the loop, even those passes that are cut short by a `continue`. The `continue` skips every statement in the block, but it does not cause the update statement to be skipped. The `for` statement may also be used unwisely, just like the `while` statement. If the `condstmt` portion of the `for` clause is omitted, it is as if a `true` conditional had been used, so that something within the `for` block must force looping to terminate. You will occasionally see the construction `for(;;)`, which is identical in meaning to `while (true)`. The two semicolons are mandatory.

The `for` statement also has some unique features that are not shared by `while`. The first is that variables may actually be declared and initialized within the `initstmt` portion. In listing 2.7 we could have dispensed with the external declaration of `x`, and put `var x = 1;` as the initialization portion of the `for` loop. This is often very convenient, since the loop variable (`x` in this case) is often used only within the loop itself, often making an external declaration pointless.

NOTE

If a variable is only used inside a block of statements it should be declared at the top of that block. This clarifies your code, since it shows which sections of code use which variables (known as *variable scope*).

A second useful feature of the `for` statement is that both the initialization portion and the update portion of the `for` clause may contain multiple statements separated by the comma operator (,). Listing 2.8 shows another version of the code in listing 2.6, rewritten so that both `x` and `lcnt` become loop variables.

Listing 2.8 A *for* Loop with Multiple Initialization and Update Statements

```

var xsum = 0;
```

```

for ( var x = 1, lcnt = 0; lcnt < 100; x++,
    lcnt++ ) {
    if ( ( x % 5 ) == 0 )           // if x is
divisible by 5
        continue;                // skip it
    xsum += x;                    // otherwise, add x
    to xsum
}

```

This usage underlines the fact that both `x` and `lcnt` are used only within the body of the `for` loop. It is also much more compact than its counterpart in listing 2.6. In this example, we need not worry about the logical effect of the `continue`; we know that both `x++` and `lcnt++` will always be executed. This is also the most common and useful way to use the comma operator.

Finally, there is another form of the `for` statement that is used exclusively with objects and arrays in JavaScript: the `for...in` statement. We will see how this is used in chapter 4, "JavaScript Objects." Figure 2.5 shows the basic structure of the `for`, `while`, and `if` statements, and the use of the `break` and `continue` statements within them.

Figure 2.5 : *Control statements determine the flow of execution in JavaScript.*

Functions and Objects

The basic statements, expressions, and operators that were discussed at the beginning of this chapter are what computer scientists usually call *primitives*. Primitives are the building blocks from which more complex elements of a program are constructed. The `for`, `while`, and `if` control structures represent the next higher level of organization in JavaScript. Each of these control structures deals with blocks of code whose execution is controlled by the various conditional tests and other clauses. The `for`, `while`, and `if` statements are all block structured.

Functions and objects represent the highest level of organization within the JavaScript language. We will spend many chapters learning how to make effective use of these concepts. The purpose of this section is to introduce them and describe their basic features.

Functions

A *function* is a block of code that has a name. Whenever that name is used the function is called, which means that the code within that function is executed. Functions may also be called with values, known as parameters, which may be used inside the body of the function. Functions serve two purposes. A function is an organizational tool, in the sense that it permits you to perform the same operation without simply copying the same code.

The second purpose of JavaScript functions is to link actions on a Web page with JavaScript code. Mouse clicks, button presses, text selections, and other user actions can call JavaScript functions by including suitable tags in the HTML source for the page.

The syntax for a function statement in JavaScript is as follows:

```
function Name ( listofparams ) {
    body
}
```

The function's `Name` is given immediately after the `function` keyword. All function names should be unique, and also should not conflict with any of the statement names which JavaScript itself uses (known as the *reserved words*). You cannot have a function named `while`, for example, and you should not have two functions both named `UserHelp`. The `listofparams` is a comma-separated list of the values that are passed into the function. These are referred to as the function's *parameters*, or *arguments*. This list may be empty, indicating that the function does not use any arguments (often called a *void function*). The function's body is the set of statements that make up the function. Listing 2.9 shows a function that adds up all the integers starting at 1 and ending at a value given as the sole argument.

Listing 2.9 A Summation function

```
function summation ( endval ) {
    var thesum = 0;           // this
    variable will hold the sum
    for ( var iter = 1; iter < endval; iter++ )
    {
        thesum += iter;      // add the
        integer into the sum
    }                        // end of the for
    loop
    return( thesum );        // return the sum
}
```

This function does the same task that came up in the discussions of the `while` and `for` statements earlier in this chapter. Now that it has been written as a function, this code never needs to be repeated again. Any time you wish to form the sum $1 + 2 + \dots + N$, you can simply call the function, as `summation(N)`, and it will perform the task. Notice that the `endval` parameter is used as the argument to the function.

When the function is called, as `summation(14)` for example, the actual value 14 is used for `endval` within the function. The function then executes the `for` statement, with `iter < 14` as its termination condition, adding in each successive value into the variable `thesum`. When the `for` loop is done, the function executes the `return` statement. This causes the

function to give the value inside the `return` statement back to the caller. This means that if we write

```
var sum14;
sum14 = summation(14);
```

the variable `sum14` is set to the value returned by the `summation` function when `endval` is given the value 14, namely 105. Functions can return any type of value, and are not restricted to returning integers.

There are several things to notice about this example. First of all, the variables `thesum` and `iter`, which are declared within the body of this function, are local variables. This means that they are only known within the body of this function, and are therefore completely unknown outside it. It is quite possible, even likely, that there are many functions, all of which have a local variable named `iter`. All these various `iters` are unrelated. Changing the value of one of these `iters` would not affect any of the others. This is why the `return` statement is necessary; it is the only way to communicate the work of the function back to the caller.

This same restriction applies to the parameter `endval` as well. The arguments to a function may *not* be changed within that function. We could well have written `endval = 15` just before the `return` statement in listing 2.9. This statement would do nothing; it certainly would not change the caller's 14 into a 15. It might seem like every function would always have a `return` statement. This is not the case, however, since it is possible for a function to have side effects without actually returning a value. This happens by referencing external objects, which are our next topic.

Objects

Functions are used to provide a uniform method for organizing code. Objects serve the same purpose for data. Up to this point, the only data items we have seen are simple variables declared with `var`. Each of these typeless quantities can only hold a single value of some sort at a time. Objects provide the ability to hold multiple values, so that a group of related data elements can be associated with one another.

What JavaScript calls an object is called a data structure (or class) in many other languages. As with JavaScript functions, there are two aspects to JavaScript objects: creating them and using them. For the moment, we will defer the question of how to create objects and concentrate on how they are used. We will also see that a JavaScript capable browser will provide a number of its own, *built-in objects*.

A JavaScript object is made up of a set of component parts, which are called its *properties*, or *members*. Suppose you have an object named `appt` which you are using to organize your appointments. The appointment object might have properties that specify the date and time of the appointment, as well as the name of the person with whom the appointment will take

place. It might also have a general description field to remind you of the purpose of this meeting. Thus, you can imagine that the `appt` object will have the following properties:

- `day`
- `month`
- `time`
- `who`
- `why`

Each of the properties of the `appt` object are referenced using the dot operator (`.`). Thus, `appt.month` refers to the `month` property and `appt.why` gives us the reason for the appointment. These references may appear on both the right and left sides of an expression; we may get their values and also set them. Listing 2.10 shows a code fragment that tests the value of `appt` and displays a message about a current appointment.

Listing 2.10 Using the *appt* Object

```
if ( appt.day == Today ) {
    document.write('<BR>You have an appointment
today<BR>');
    document.write('See ' + appt.who + ' at ' +
appt.time<BR>');
    document.write(appt.why + '<BR>');
}
```

This example assumes that the variable `Today` has somehow been initialized with today's date, so that the equality test with `appt.day` is only true for today's appointments. If the test does succeed then the three statements in the `if` block are executed. Each of these references `document.write`. The `document` object is a built-in object of the Netscape Navigator browser. This object has a member known as `write`, which is actually a function. Functional members of JavaScript objects are known as *methods*. This particular method takes a string and displays it on the current Web page.

Each of the three strings that are passed to `document.write` are constructed using `+` as a string concatenation operator. Each of them references one or more properties of the `appt` object in order to provide meaningful messages to the user. Each also includes `
`, the HTML construction for a line break. This ability to directly issue HTML directives is one of the most powerful aspects of JavaScript, as it allows the programmer to dynamically modify the contents of Web pages using JavaScript functions and objects.

Once you learn more about the `Date` object, in "The *Date* Object" section of chapter 5, you will be able to construct a much more satisfying version of this example. Even at this stage, however, the advantage of object-based programming should be apparent. Rather than carrying about many variables, you can use objects instead. Each object can contain all the variables of interest to a particular idea. It can also contain method functions that perform

related work. Objects can even contain other objects, so that you can organize your data in a hierarchical structure. Subsequent chapters explore these ideas in much greater detail.

MySQL Reference Manual

Copyright © 1997, 1998, 1999 TcX AB, Detron HB and Monty Program KB

..

1 General Information about MySQL

This is the **MySQL** reference manual; it documents **MySQL** version 3.23.9-alpha.

MySQL is a very fast, multi-threaded, multi-user and robust SQL (Structured Query Language) database server.

For Unix and OS/2 platforms, **MySQL** is basically free; for Microsoft platforms you must get a **MySQL** license after a trial time of 30 days. See Chapter 3 [Licensing and Support], page 24.

The **MySQL** home page (<http://www.mysql.com/>) provides the latest information about **MySQL**.

For a discussion of **MySQL**'s capabilities, see Section 1.5 [Features], page 5.

For installation instructions, see Chapter 4 [Installing], page 34. For tips on porting **MySQL** to new machines or operating systems, see Appendix G [Porting], page 492.

For information about upgrading from a 3.21 release, see Section 4.16.2 [Upgrading-from-3.21], page 93.

For a tutorial introduction to **MySQL**, see Chapter 8 [Tutorial], page 229.

For examples of SQL and benchmarking information, see the benchmarking directory ('sql-bench' in the distribution).

For a history of new features and bug fixes, see Appendix D [News], page 441.

For a list of currently known bugs and misfeatures, see Appendix E [Bugs], page 486.

For future plans, see Appendix F [TODO], page 488.

For a list of all the contributors to this project, see Appendix C [Credits], page 435.

IMPORTANT:

Send error (often called bugs) reports, questions and comments to the mailing list at mysql@lists.mysql.com. See Section 2.3 [Bug reports], page 18.

For source distributions, the `mysqlbug` script can be found in the 'scripts' directory. For binary distributions, `mysqlbug` can be found in the 'bin' directory.

If you have any suggestions concerning additions or corrections to this manual, please send them to the manual team at (docs@mysql.com).

1.1 What is MySQL?

MySQL is a true multi-user, multi-threaded SQL database server. SQL (Structured Query Language) is the most popular and standardized database language in the world. **MySQL** is a client/server implementation that consists of a server daemon `mysqld` and many different client programs and libraries.

SQL is a standardized language that makes it easy to store, update and access information. For example, you can use SQL to retrieve product information and store customer information for a web site. **MySQL** is also fast and flexible enough to allow you to store logs and pictures in it.

The main goals of **MySQL** are speed, robustness and ease of use. **MySQL** was originally developed because we needed a SQL server that could handle very large databases an order of magnitude faster than what any database vendor could offer to us on inexpensive hardware. We have now been using **MySQL** since 1996 in an environment with more than 40 databases containing 10,000 tables, of which more than 500 have more than 7 million rows. This is about 100 gigabytes of mission-critical data.

The base upon which **MySQL** is built is a set of routines that have been used in a highly demanding production environment for many years. Although **MySQL** is still under development, it already offers a rich and highly useful function set.

The official way to pronounce **MySQL** is “My Ess Que Ell” (Not MY-*SE*QUEL).

1.2 About this manual

This manual is currently available in Texinfo, plain text, Info, HTML, PostScript and PDF versions. Because of their size, PostScript and PDF versions are not included with the main **MySQL** distribution, but are available for separate download at <http://www.mysql.com>.

The primary document is the Texinfo file. The HTML version is produced automatically with a modified version of `texi2html`. The plain text and Info versions are produced with `makeinfo`. The Postscript version is produced using `texi2dvi` and `dvips`. The PDF version is produced with `pdftex`.

This manual is written and maintained by David Axmark, Michael (Monty) Widenius and Paul DuBois. For other contributors, see Appendix C [Credits], page 435.

1.2.1 Conventions used in this manual

This manual uses certain typographical conventions:

constant Constant-width font is used for command names and options; SQL statements; database, table and column names; C and Perl code; and environment variables. Example: “To see how `mysqladmin` works, invoke it with the `--help` option.”

`'filename'`

Constant-width font with surrounding quotes is used for filenames and pathnames. Example: “The distribution is installed under the `'/usr/local/'` directory.”

`'c'`

Constant-width font with surrounding quotes is also used to indicate character sequences. Example: “To specify a wildcard, use the `'%'` character.”

italic

Italic font is used for emphasis, *like this*.

boldface

Boldface font is used for access privilege names (e.g., “do not grant the **process** privilege lightly”) and to convey **especially strong emphasis**.

When commands are shown that are meant to be executed by a particular program, the program is indicated by the prompt shown with the command. For example, `shell>` indicates a command that you execute from your login shell, and `mysql>` indicates a command that you execute from the `mysql` client:

```
shell> type a shell command here
mysql> type a mysql command here
```

Shell commands are shown using Bourne shell syntax. If you are using a csh-style shell, you may need to issue commands slightly differently. For example, the sequence to set an environment variable and run a command looks like this in Bourne shell syntax:

```
shell> VARNAME=value some_command
```

For csh, you would execute the sequence like this:

```
shell> setenv VARNAME value
shell> some_command
```

Database, table and column names often must be substituted into commands. To indicate that such substitution is necessary, this manual uses `db_name`, `tbl_name` and `col_name`. For example, you might see a statement like this:

```
mysql> SELECT col_name FROM db_name.tbl_name;
```

This means that if you were to enter a similar statement, you would supply your own database, table and column names, perhaps like this:

```
mysql> SELECT author_name FROM biblio_db.author_list;
```

SQL statements may be written in uppercase or lowercase. When this manual shows a SQL statement, uppercase is used for particular keywords if those keywords are under discussion (to emphasize them) and lowercase is used for the rest of the statement. So you might see the following in a discussion of the `SELECT` statement:

```
mysql> SELECT count(*) FROM tbl_name;
```

On the other hand, in a discussion of the `COUNT()` function, the statement would be written like this:

```
mysql> select COUNT(*) from tbl_name;
```

If no particular emphasis is intended, all keywords are written uniformly in uppercase.

In syntax descriptions, square brackets (`'[`' and `']`') are used to indicate optional words or clauses:

```
DROP TABLE [IF EXISTS] tbl_name
```

When a syntax element consists of a number of alternatives, the alternatives are separated by vertical bars (`'|`'). When one member from a set of choices may be chosen, the alternatives are listed within square brackets. When one member from a set of choices must be chosen, the alternatives are listed within braces (`'{'` and `'}'`):

```
TRIM([[BOTH | LEADING | TRAILING] [remstr] FROM] str)
{DESCRIBE | DESC} tbl_name {col_name | wild}
```

1.3 History of MySQL

We once started off with the intention of using `mSQL` to connect to our tables using our own fast low-level (ISAM) routines. However, after some testing we came to the conclusion that `mSQL` was not fast enough or flexible enough for our needs. This resulted in a new SQL interface to our database but with almost the same API interface as `mSQL`. This API was chosen to ease porting of third-party code.

The derivation of the name **MySQL** is not perfectly clear. Our base directory and a large number of our libraries and tools have had the prefix “my” for well over 10 years. However, Monty’s daughter (some years younger) is also named My. So which of the two gave its name to **MySQL** is still a mystery, even for us.

1.4 Books about MySQL

While this manual is still the right place for up to date technical information, its primary goal is to contain everything there is to know about **MySQL**. And it is sometimes nice to have a bound book to read in bed or while you travel. Here are a list of books about MySQL (in English).

Title	MySQL (http://www2.newriders.com/cfm/prod_book.cfm?RecordID=584)
Publisher	New Riders
Author	Paul DuBois
Pub Date	1st Edition December 1999
ISBN	0735709211
Pages	800
Price	\$49.99 US
Downloadable examples	samp_db.tar.gz (http://www.mysql.com/Contrib/Examples/samp_db.tar.gz)

Foreword by Michael “Monty” Widenius, MySQL Moderator

In MySQL, Paul DuBois provides you with a comprehensive guide to one of the most popular relational database systems. Paul has contributed to the online documentation for MySQL, and is an active member of the MySQL community. The principal MySQL developer, Monty Widenius, and a network of his fellow developers reviewed the manuscript, providing Paul with the kind of insight no one else could supply.

Instead of merely giving you a general overview of MySQL, Paul teaches you how to make the most of its capabilities. Through two sample database applications that run throughout the book, he gives you solutions to problems you’re sure to face. He helps you integrate MySQL efficiently with third-party tools, such as PHP and Perl, enabling you to generate dynamic Web pages through database queries. He teaches you to write programs that access MySQL databases, and also provides a comprehensive set of references to column types, operators, functions, SQL syntax, MySQL programming, C API, Perl DBI, and PHP API. MySQL simply gives you the kind of information you won’t find anywhere else.

If you use MySQL, this book provides you with:

- An introduction to MySQL and SQL
- Coverage of MySQL’s data types and how to use them
- Thorough treatment of how to write client programs in C
- A guide to using the Perl DBI and PHP APIs for developing command-line and Web-based applications
- Tips on administrative issues such as user accounts, backup, crash recovery, and security

- A privilege and password system which is very flexible and secure, and which allows host-based verification. Passwords are secure since all password traffic when connecting to a server is encrypted.
- ODBC (Open-DataBase-Connectivity) for Windows95 (with source). All ODBC 2.5 functions and many others. You can, for example, use Access to connect to your **MySQL** server. See Chapter 16 [ODBC], page 337.
- Very fast B-tree disk tables with index compression.
- 16 indexes per table are allowed. Each index may consist of 1 to 16 columns or parts of columns. The maximum index length is 256 bytes (this may be changed when compiling **MySQL**). An index may use a prefix of a **CHAR** or **VARCHAR** field.
- Fixed-length and variable-length records.
- In-memory hash tables which are used as temporary tables.
- Handles large databases. We are using **MySQL** with some databases that contain 50,000,000 records.
- All columns have default values. You can use **INSERT** to insert a subset of a table's columns; those columns that are not explicitly given values are set to their default values.
- Uses GNU Automake, Autoconf, and libtool for portability.
- Written in C and C++. Tested with a broad range of different compilers.
- A very fast thread-based memory allocation system.
- No memory leaks. Tested with a commercial memory leakage detector (purify).
- Includes **myisamchk**, a very fast utility for table checking, optimization and repair. See Chapter 13 [Maintenance], page 315.
- Full support for the ISO-8859-1 Latin1 character set. For example, the Scandinavian characters å, ä and ö are allowed in table and column names.
- All data are saved in ISO-8859-1 Latin1 format. All comparisons for normal string columns are case insensitive.
- Sorting is done according to the ISO-8859-1 Latin1 character set (the Swedish way at the moment). It is possible to change this in the source by adding new sort order arrays. To see an example of very advanced sorting, look at the Czech sorting code. **MySQL** supports many different character sets that can be specified at compile time.
- Aliases on tables and columns as in the SQL92 standard.
- **DELETE**, **INSERT**, **REPLACE**, and **UPDATE** return how many rows were changed (affected).
- Function names do not clash with table or column names. For example, **ABS** is a valid column name. The only restriction is that for a function call, no spaces are allowed between the function name and the '(' that follows it. See Section 7.31 [Reserved words], page 226.
- All **MySQL** programs can be invoked with the **--help** or **-?** options to obtain online assistance.
- The server can provide error messages to clients in many languages. See Section 9.1 [Languages], page 262.

- Help in choosing an ISP for MySQL access
- A comprehensive reference for MySQL's data types, operators, functions, and SQL statements and utilities
- Complete reference guides for MySQL's C API, the Perl DBI API, and PHP's MySQL-related functions

Title	MySQL & mSQL (http://www.oreilly.com/catalog/msql/noframes.htm)
Publisher	O'Reilly
Authors	Randy Jay Yarger, George Reese & Tim King
Pub Date	1st Edition July 1999
ISBN	1-56592-434-7, Order Number: 4347
Pages	506
Price	\$34.95

This book teaches you how to use MySQL and mSQL, two popular and robust database products that support key subsets of SQL on both Linux and UNIX systems. Anyone who knows basic C, Java, Perl, or Python can write a program to interact with a database, either as a stand-alone application or through a Web page. This book takes you through the whole process, from installation and configuration to programming interfaces and basic administration. Includes ample tutorial material.

1.5 The main features of MySQL

The following list describes some of the important characteristics of MySQL:

- Fully multi-threaded using kernel threads. That means it easily can use multiple CPUs if available.
- C, C++, Eiffel, Java, Perl, PHP, Python and TCL APIs. See Chapter 20 [Clients], page 362.
- Works on many different platforms. See Section 4.2 [Which OS], page 36.
- Many column types: signed/unsigned integers 1, 2, 3, 4 and 8 bytes long, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET and ENUM types. See Section 7.3 [Column types], page 133.
- Very fast joins using an optimized one-sweep multi-join.
- Full operator and function support in the SELECT and WHERE parts of queries. Example:

```
mysql> SELECT CONCAT(first_name, " ", last_name) FROM tbl_name
        WHERE income/dependents > 10000 AND age > 30;
```
- SQL functions are implemented through a highly-optimized class library and should be as fast as they can get! Usually there shouldn't be any memory allocation at all after query initialization.
- Full support for SQL GROUP BY and ORDER BY clauses. Support for group functions (COUNT(), COUNT(DISTINCT), AVG(), STD(), SUM(), MAX() and MIN()).
- Support for LEFT OUTER JOIN with ANSI SQL and ODBC syntax.
- You can mix tables from different databases in the same query (as of version 3.22).

- Clients connect to the **MySQL** server using TCP/IP connections or Unix sockets, or named pipes under NT.
- The **MySQL**-specific **SHOW** command can be used to retrieve information about databases, tables and indexes. The **EXPLAIN** command can be used to determine how the optimizer resolves a query.

1.6 How stable is MySQL?

This section addresses the questions, “how stable is **MySQL**?” and, “can I depend on **MySQL** in this project?” Here we will try to clarify some issues and to answer some of the more important questions that seem to concern many people. This section has been put together from information gathered from the mailing list (which is very active in reporting bugs).

At TcX, **MySQL** has worked without any problems in our projects since mid-1996. When **MySQL** was released to a wider public, we noticed that there were some pieces of “untested code” that were quickly found by the new users who made queries in a manner different than our own. Each new release has had fewer portability problems than the previous one (even though each has had many new features), and we hope that it will be possible to label one of the next releases “stable”.

Each release of **MySQL** has been usable and there have been problems only when users start to use code from “the gray zones”. Naturally, outside users can’t know what the gray zones are; this section attempts to indicate those that are currently known. The descriptions deal with the 3.22.x version of **MySQL**. All known and reported bugs are fixed in the latest version, with the exception of the bugs listed in the bugs section, which are things that are “design”-related. See Appendix E [Bugs], page 486.

MySQL is written in multiple layers and different independent modules. These modules are listed below with an indication of how well-tested each of them is:

The ISAM table handler — Stable

This manages storage and retrieval of all data in **MySQL** 3.22 and earlier versions. In all **MySQL** releases there hasn’t been a single (reported) bug in this code. The only known way to get a corrupted table is to kill the server in the middle of an update. Even that is unlikely to destroy any data beyond rescue, because all data are flushed to disk between each query. There hasn’t been a single bug report about lost data because of bugs in **MySQL**, either.

The MyISAM table handler — Beta

This is new in **MySQL** 3.23. It’s largely based on the ISAM table code but has a lot of new very useful features.

The parser and lexical analyser — Stable

There hasn’t been a single reported bug in this system for a long time.

The C client code — Stable

No known problems. In early 3.20 releases, there were some limitations in the send/receive buffer size. As of 3.21.x, the buffer size is now dynamic up to a default of 24M.

Standard client programs — Stable

These include `mysql`, `mysqladmin` and `mysqlshow`, `mysqldump`, and `mysqlimport`.

Basic SQL — Stable

The basic SQL function system and string classes and dynamic memory handling. Not a single reported bug in this system.

Query optimizer — Stable

Range optimizer — Gamma

Join optimizer — Stable

Locking — Gamma

This is very system-dependent. On some systems there are big problems using standard OS locking (`fcntl()`). In these cases, you should run the **MySQL** daemon with the `--skip-locking` flag. Problems are known to occur on some Linux systems and on SunOS when using NFS-mounted file systems.

Linux threads — Gamma

The only problem found has been with the `fcntl()` call, which is fixed by using the `--skip-locking` option to `mysqld`. Some people have reported lockup problems with the 0.5 release.

Solaris 2.5+ pthreads — Stable

We use this for all our production work.

MIT-pthreads (Other systems) — Gamma

There have been no reported bugs since 3.20.15 and no known bugs since 3.20.16. On some systems, there is a “misfeature” where some operations are quite slow (a 1/20 second sleep is done between each query). Of course, MIT-pthreads may slow down everything a bit, but index-based `SELECT` statements are usually done in one time frame so there shouldn’t be a mutex locking/thread juggling.

Other thread implementations — Alpha - Beta

The ports to other systems are still very new and may have bugs, possibly in **MySQL**, but most often in the thread implementation itself.

LOAD DATA ..., INSERT ... SELECT — Stable

Some people have thought they have found bugs here, but these usually have turned out to be misunderstandings. Please check the manual before reporting problems!

ALTER TABLE — Stable

Small changes in 3.22.12.

DBD — Stable

Now maintained by Jochen Wiedmann wiedmann@neckar-alb.de. Thanks!

mysqlaccess — Stable

Written and maintained by Yves Carlier Yves.Carlier@rug.ac.be. Thanks!

GRANT — Gamma

Big changes made in **MySQL** 3.22.12.

MyODBC (uses ODBC SDK 2.5) — Gamma

It seems to work well with some programs.

TcX provides email support for paying customers, but the **MySQL** mailing list usually provides answers to common questions. Bugs are usually fixed right away with a patch; for serious bugs, there is almost always a new release.

1.7 Year 2000 compliance

MySQL itself has no problems with Year 2000 (Y2K) compliance:

- **MySQL** uses Unix time functions and has no problems with dates until 2069; all 2-digit years are regarded to be in the range 1970 to 2069, which means that if you store 01 in a year column, **MySQL** treats it as 2001.
- All **MySQL** date functions are stored in one file 'sql/time.cc' and coded very carefully to be year 2000-safe.
- In **MySQL** 3.22 and later versions, the new **YEAR** column type can store years 0 and 1901 to 2155 in 1 byte and display them using 2 or 4 digits.

You may run into problems with applications that use **MySQL** in a way that is not Y2K-safe. For example, many old applications store or manipulate years using 2-digit values (which are ambiguous) rather than 4-digit values. This problem may be compounded by applications that use values such as 00 or 99 as "missing" value indicators.

Unfortunately, these problems may be difficult to fix, since different applications may be written by different programmers, each of whom may use a different set of conventions and date-handling functions.

Here is a simple demonstration illustrating that **MySQL** doesn't have any problems with dates until the year 2030!

```
mysql> DROP TABLE IF EXISTS y2k;
mysql> CREATE TABLE y2k (date date, date_time datetime, time_stamp timestamp);
mysql> INSERT INTO y2k VALUES ("1998-12-31","1998-12-31 23:59:59",19981231235959);
mysql> INSERT INTO y2k VALUES ("1999-01-01","1999-01-01 00:00:00",19990101000000);
mysql> INSERT INTO y2k VALUES ("1999-09-09","1999-09-09 23:59:59",19990909235959);
mysql> INSERT INTO y2k VALUES ("2000-01-01","2000-01-01 00:00:00",20000101000000);
mysql> INSERT INTO y2k VALUES ("2000-02-28","2000-02-28 00:00:00",20000228000000);
mysql> INSERT INTO y2k VALUES ("2000-02-29","2000-02-29 00:00:00",20000229000000);
mysql> INSERT INTO y2k VALUES ("2000-03-01","2000-03-01 00:00:00",20000301000000);
mysql> INSERT INTO y2k VALUES ("2000-12-31","2000-12-31 23:59:59",20001231235959);
mysql> INSERT INTO y2k VALUES ("2001-01-01","2001-01-01 00:00:00",20010101000000);
mysql> INSERT INTO y2k VALUES ("2004-12-31","2004-12-31 23:59:59",20041231235959);
mysql> INSERT INTO y2k VALUES ("2005-01-01","2005-01-01 00:00:00",20050101000000);
mysql> INSERT INTO y2k VALUES ("2030-01-01","2030-01-01 00:00:00",20300101000000);
mysql> INSERT INTO y2k VALUES ("2050-01-01","2050-01-01 00:00:00",20500101000000);
mysql> SELECT * FROM y2k;
```

date	date_time	time_stamp
1998-12-31	1998-12-31 23:59:59	19981231235959
1999-01-01	1999-01-01 00:00:00	19990101000000
1999-09-09	1999-09-09 23:59:59	19990909235959
2000-01-01	2000-01-01 00:00:00	20000101000000
2000-02-28	2000-02-28 00:00:00	20000228000000
2000-02-29	2000-02-29 00:00:00	20000229000000
2000-03-01	2000-03-01 00:00:00	20000301000000
2000-12-31	2000-12-31 23:59:59	20001231235959
2001-01-01	2001-01-01 00:00:00	20010101000000
2004-12-31	2004-12-31 23:59:59	20041231235959
2005-01-01	2005-01-01 00:00:00	20050101000000
2030-01-01	2030-01-01 00:00:00	20300101000000
2050-01-01	2050-01-01 00:00:00	20500101000000

```

+-----+-----+-----+
| 1998-12-31 | 1998-12-31 23:59:59 | 19981231235959 |
| 1999-01-01 | 1999-01-01 00:00:00 | 19990101000000 |
| 1999-09-09 | 1999-09-09 23:59:59 | 19990909235959 |
| 2000-01-01 | 2000-01-01 00:00:00 | 20000101000000 |
| 2000-02-28 | 2000-02-28 00:00:00 | 20000228000000 |
| 2000-02-29 | 2000-02-29 00:00:00 | 20000229000000 |
| 2000-03-01 | 2000-03-01 00:00:00 | 20000301000000 |
| 2000-12-31 | 2000-12-31 23:59:59 | 20001231235959 |
| 2001-01-01 | 2001-01-01 00:00:00 | 20010101000000 |
| 2004-12-31 | 2004-12-31 23:59:59 | 20041231235959 |
| 2005-01-01 | 2005-01-01 00:00:00 | 20050101000000 |
| 2030-01-01 | 2030-01-01 00:00:00 | 20300101000000 |
| 2050-01-01 | 2050-01-01 00:00:00 | 00000000000000 |
+-----+-----+-----+

```

13 rows in set (0.00 sec)

This shows that the **DATE** and **DATETIME** types are will not give any problems with future dates (they handle dates until the year 9999).

The **TIMESTAMP** type, that is used to store the current time, has a range up to only 2030-01-01. **TIMESTAMP** has a range of 1970 to 2030 on 32-bit machines (signed value). On 64-bit machines it handles times up to 2106 (unsigned value).

Even though **MySQL** is Y2K-compliant, it is your responsibility to provide unambiguous input. See Section 7.3.3.1 [Y2K issues], page 141 for **MySQL**'s rules for dealing with ambiguous date input data (data containing 2-digit year values).

1.8 General SQL information and tutorials

This book has been recommended by a several people on the **MySQL** mailing list:

Judith S. Bowman, Sandra L. Emerson and Marcy Darnovsky
 The Practical SQL Handbook: Using Structured Query Language
 Second Edition
 Addison-Wesley
 ISBN 0-201-62623-3
<http://www.awl.com>

This book has also received some recommendations by **MySQL** users:

Martin Gruber
 Understanding SQL
 ISBN 0-89588-644-8
 Publisher Sybex 510 523 8233
 Alameda, CA USA

A SQL tutorial is available on the net at <http://www.geocities.com/SiliconValley/Vista/2207/sql1>.

SQL in 21 Tagen (online book in German language): <http://www.mut.de/lesecke/buecher/sql/inhal>

1.9 Useful MySQL-related links

Tutorials

- A beginner's tutorial of how to start using **MySQL** (<http://www.devshed.com/resource/advanced/my>)
- <http://www.analysisandsolutions.com/code/mybasic.htm> Beginners **MySQL** Tutorial on how to install and set up **MySQL** on a Windows machine.
- A lot of **MySQL** tutorials (http://www.devshed.com/Server_Side/MySQL/)
- Setting Up a MySQL Based Website (<http://www.linuxplanet.com/linuxplanet/tutorials/1046/>)
- **MySQL**-perl tutorial (<http://www.hotwired.com/webmonkey/backend/tutorials/tutorial1.html>)
- PHP/MySQL Tutorial (<http://www.hotwired.com/webmonkey/databases/tutorials/tutorial4.b>)

Porting MySQL / Using MySQL on different systems

- The MacOS Xclave (<http://xclave.macnn.com/MySQL/>). Running MySQL on MacOSX
- MySql for MacOSX Server (<http://www.prnet.de/RegEx/mysql.html>)
- Client libraries for the Macintosh (<http://www.lilback.com/macsql/>)

Perl related links

- Perl DBI with MySQL FAQ (http://haven.e-cactus.com/dbi_mysql)

MySQL discussion forums

- Examples using **MySQL**; (check Top 20) (<http://webdev.weberdev.com/>)

Commercial applications that support MySQL

- SupportWizard; Interactive helpdesk on the web (This product includes a licensed copy of MySQL) (<http://www.supportwizard.com/>)
- Right Now Web; Web automation for customer service (<http://www.rightnowtech.com/>)
- Bazaar; Interactive Discussion Forums with web interface (<http://www.icaap.org/Bazaar/>)
- PhoneSweepT (<http://www.phonesweep.com/>) is the world's first commercial Telephone Scanner. Many break-ins in recent years have come not through the Internet, but through unauthorized dial-up modems. PhoneSweep lets you find these modems by repeatedly placing phone calls to every phone number that your organization controls. PhoneSweep has a built-in expert system that can recognize more than 250 different kinds of remote-access programs, including Carbon CopyT, pcANYWHERE, and Windows NT RAS. All information is stored in the SQL database. It then generates a comprehensive report detailing which services were discovered on which dial-up numbers in your organization.

SQL Clients/Report writers

- **MySQL Editor/Utility** for MS Windows Platforms. (<http://www.urbanresearch.com/software/util>)
- **KDE MySQL client** (<http://www.xnot.com/kmysql>)
- **KMySQL** (<http://www.penguinpowered.com/~kmysql>) KMySQL is a database client for KDE that primarily supports **MySQL**.
- **Kiosk**; a MySQL client for database management (<http://www.icaap.org/software/kiosk/>). Written in Perl. Will be a part of Bazaar.
- A free report writer in Java (<http://www.geocities.com/SiliconValley/Ridge/4280/GenericRepo>).

Web development tools that support MySQL

- **PHP**: A server-side HTML-embedded scripting language (<http://www.php.net/>)
- **The Midgard Application Server**; a powerful Web development environment based on MySQL and PHP (<http://www.midgard-project.org>)
- **SmartWorker** is a platform for web application development (<http://www.smartworker.org>)
- **XSP**: e(X)tensible (s)erver (p)ages and is a HTML embedded tag language written in Java (previously known as XTAGS) (<http://xsp.lentus.se/>)
- **dbServ** (<http://www.dbServ.de/>) is an extension to a web server to integrate databases output into your HTML code. You may use any HTML function in your output. Only the client will stop you. It works as standalone server or as JAVA servlet.
- **Platform independent ASP** from ChiliSoft (<http://www.chilisoft.com/>)
- **MySQL + PHP demos** (<http://www.wernhart.priv.at/php/>)
- **ForwardSQL**: HTML interface to manipulate **MySQL** databases (<http://www.dbwww.com/>)
- **WWW-SQL**: Display database information (<http://www.daa.com.au/~james/www-sql/>)
- **Minivend**: A Web shopping cart (<http://www.minivend.com/minivend/>)
- **HeiTML**: A server-side extension of HTML and a 4GL language at the same time (<http://www.heitml.com/>)
- **Metahtml**: A Dynamic Programming Language for WWW Applications (<http://www.metahtml.com/>)
- **VelocityGen** for Perl and TCL (<http://www.binevolve.com/>)
- **Hawkeye Internet Server Suite** (<http://hawkeye.net/>)
- **Network Database Connection For Linux** (<http://www.fastflow.com/>)
- **WDBI**: Web browser as a universal front end to databases which supports **MySQL** well. (<http://www.wdbi.net/>)
- **WebGroove Script**: HTML compiler and server-side scripting language (<http://www.webgroove.com/>)
- A server-side web site scripting language (<http://www.ihtml.com/>)
- How to use **MySQL** with ColdFusion on Solaris (<ftp://ftp.igc.apc.org/pub/myodbc/README>)
- **Calistra's ODBC MySQL Administrator** (<http://calistra.com/MySQL/>)
- **Webmerger** (<http://www.webmerger.com>) This CGI tool interprets files and generates dynamic output based on a set of simple tags. Ready-to-run drivers for **MySQL** and PostgreSQL through ODBC.

- PHPclub (http://phpclub.unet.ru/index_e.php3). Tips and tricks for PHP
- MySQL and Perl Scripts (<http://www.penguinservices.com/scripts>)
- The Widgetchuck; Web Site Tools and Gadgets (<http://www.widgetchuck.com>)
- AdCycle (<http://www.adcycle.com/>) advertising management software
- pwPage (<http://www.bidsystems.com/pwPage>) provides an extremely fast and simple approach to the creation of data base forms. That is, if a data base table exists and an HTML page has been constructed using a few simple guidelines, pwPage can be immediately used for table data selections, insertions, updates, deletions and selectable table content reviewing.

Database design tools with MySQL support

- "Dezign for databases" is a database development tool using an rick> entity relationship diagram (ERD). (<http://www.heraut.demon.nl/dezign/dezign.html>)

Web servers with MySQL tools

- An Apache authentication module (http://bourbon.netvision.net.il/mysql/mod_auth_mysql/)
- The Roxen Challenger Web server (<http://www.roxen.com/>)

Extensions for other programs

A Delphi interface to MySQL. (<http://www.fichtner.net/delphi/mysql.delphi.phtml>)
With source code. By Matthias Fichtner.

- TMySQL; A library to use MySQL with Delphi (<http://www.productivity.org/projects/mysql/>)
- Delphi TDataset-component (<http://www.geocities.com/CapeCanaveral/2064/mysql.html>)
- MySQL support for BIND (The Internet Domain Name Server) (http://www.seawood.org/msql_bind/)
- MySQL support for Sendmail and Procmail (<http://www.inet-interactive.com/sendmail>)

Using MySQL with other programs

- Using MySQL with Access (<http://www.iserver.com/support/addonhelp/database/mysql/msacce>)

ODBC related links

- Popular iODBC Driver Manager (libiodbc) now available in Open Source format (<http://www.iodbc.org/>)
- The FreeODBC Pages (<http://users.ids.net/~bjepson/freeODBC/>)
- unixodbc (<http://http://genix.net/unixODBC/>) The unixODBC Project goals are to develop and promote unixODBC to be the definitive standard for ODBC on the Linux platform. This is to include GUI support for KDE.

API related links

- www.jpopp.com (<http://www.jpopp.com>) Partially implemented TDataset-compatible components for **MySQL**.
- <http://www.riverstyx.net/qpopmysql/>) A patch to allow POP3 authentication from a **MySQL** database. There's also a link to Paul Khavkine's patch for Procmail to allow any MTA to deliver to users in a **MySQL** database.
- Visual Basic class generator for Active X (<http://www.pbc.ottawa.on.ca>)
- Client libraries for the Macintosh (<http://www.lilback.com/macsql/>)
- **MySQL** binding to Free Pascal (<http://tfdec1.fys.kuleuven.ac.be/~michael/fpc-linux/mysql>)
- SCMDB (<http://www.dedecker.net/jessie/scmdb/>). SCMDB is an add-on for SCM that ports the mysql C library to scheme (SCM). With this library scheme developers can make connections to a **MySQL** database and use embedded SQL in their programs.

Other MySQL-related links

- Registry of Web providers who support **MySQL** (<http://www.wix.com/mysql-hosting>)
- Links about using **MySQL** in Japan/Asia (<http://www.softagency.co.jp/mysql/index.en.phtml>)
- Commercial Web defect tracking system (<http://www.open.com.au/products.html>)
- PTS: Project Tracking System (<http://www.stonekeep.com/pts/>)
- Job and software tracking system (<http://tomato.nvgc.vt.edu/~hroberts/mot>)
- ExportSQL: A script to export data from Access95+ (<http://www.cynergi.net/non-secure/exports>)
- SAL (Scientific Applications on Linux) **MySQL** entry (<http://SAL.KachinaTech.COM/H/1/MYSQL.htm>)
- A consulting company which mentions **MySQL** in the right company (<http://www.infotech-nj.com/i>)
- PMP Computer Solutions. Database developers using **MySQL** and **mSQL** (<http://www.pmpcs.com/>)
- Airborne Early Warning Association (<http://www.aewa.org>)
- **MySQL** UDF Registry (<http://abattoir.cc.ndsu.nodak.edu/~nem/mysql/udf/>)
- Y2K tester (<http://21ccs.com/~gboersm/y2kmatrix/>)

SQL and database interfaces

- The JDBC database access API (<http://java.sun.com/products/jdbc/>)
- Patch for **mSQL** TCL (<http://www.gagme.com/mysql>)
- EasySQL: An ODBC-like driver manager (<http://www.amsoft.ru/easysql/>)
- A REXX interface to SQL databases (<http://www.lightlink.com/hessling/rexxsql.html>)
- TCL interface (<http://www.binevolve.com/~tdarugar/tcl-sql>)

Examples of MySQL use

- Little6 Inc (<http://www.little6.com/about/linux/>) An online contract and job finding site that is powered by **MySQL**, PHP3 and Linux.

6.5 Connecting to the MySQL server

MySQL client programs generally require that you specify connection parameters when you want to access a MySQL server: the host you want to connect to, your user name and your password. For example, the `mysql` client can be started like this (optional arguments are enclosed between '[' and '']):

```
shell> mysql [-h host_name] [-u user_name] [-p your_pass]
```

Alternate forms of the `-h`, `-u` and `-p` options are `--host=host_name`, `--user=user_name` and `--password=your_pass`. Note that there is *no space* between `-p` or `--password=` and the password following it.

Note: Specifying a password on the command line is not secure! Any user on your system may then find out your password by typing a command like: `ps auxww`. See Section 4.15.4 [Option files], page 89.

`mysql` uses default values for connection parameters that are missing from the command line:

- The default hostname is `localhost`.
- The default user name is your Unix login name.
- No password is supplied if `-p` is missing.

Thus, for a Unix user `joe`, the following commands are equivalent:

```
shell> mysql -h localhost -u joe
shell> mysql -h localhost
shell> mysql -u joe
shell> mysql
```

Other MySQL clients behave similarly.

On Unix systems, you can specify different default values to be used when you make a connection, so that you need not enter them on the command line each time you invoke a client program. This can be done in a couple of ways:

- You can specify connection parameters in the `[client]` section of the `my.cnf` configuration file in your home directory. The relevant section of the file might look like this:

```
[client]
host=host_name
user=user_name
password=your_pass
```

See Section 4.15.4 [Option files], page 89.

- You can specify connection parameters using environment values. The host can be specified using `MYSQL_HOST`. The MySQL user name can be specified using `USER` (this is for Windows only). The password can be specified using `MYSQL_PWD` (but this is insecure; see next section).

If connection parameters are specified in multiple ways, values specified on the command line take precedence over values specified in configuration files and environment variables, and values in configuration files take precedence over values in environment variables.

```

mysql> INSERT INTO user (Host,User,Password)
VALUES('server.domain','custom',PASSWORD('stupid'));
mysql> INSERT INTO user (Host,User,Password)
VALUES('whitehouse.gov','custom',PASSWORD('stupid'));
mysql> INSERT INTO db
(Host,Db,User,Select_priv,Insert_priv,Update_priv,Delete_priv,
Create_priv,Drop_priv)
VALUES
('localhost','bankaccount','custom','Y','Y','Y','Y','Y','Y');
mysql> INSERT INTO db
(Host,Db,User,Select_priv,Insert_priv,Update_priv,Delete_priv,
Create_priv,Drop_priv)
VALUES
('whitehouse.gov','expenses','custom','Y','Y','Y','Y','Y','Y');
mysql> INSERT INTO db
(Host,Db,User,Select_priv,Insert_priv,Update_priv,Delete_priv,
Create_priv,Drop_priv)
VALUES('','customer','custom','Y','Y','Y','Y','Y','Y');
mysql> FLUSH PRIVILEGES;

```

The first three INSERT statements add user table entries that allow user custom to connect from the various hosts with the given password, but grant no permissions to him (all privileges are set to the default value of 'N'). The next three INSERT statements add db table entries that grant privileges to custom for the bankaccount, expenses and customer databases, but only when accessed from the proper hosts. As usual, when the grant tables are modified directly, the server must be told to reload them (with FLUSH PRIVILEGES) so that the privilege changes take effect.

If you want to give a specific user access from any machine in a given domain, you can issue a GRANT statement like the following:

```

mysql> GRANT ...
      ON *.*
      TO myusername@"%.mydomainname.com"
      IDENTIFIED BY 'mypassword';

```

To do the same thing by modifying the grant tables directly, do this:

```

mysql> INSERT INTO user VALUES ('%.mydomainname.com', 'myusername',
      PASSWORD('mypassword'),...);
mysql> FLUSH PRIVILEGES;

```

You can also use `xmysqladmin`, `mysql_webadmin` and even `xmysql` to insert, change and update values in the grant tables. You can find these utilities at the **MySQL Contrib** directory (<http://www.mysql.com/Contrib/>).

6.14 How to set up passwords

The examples in the preceding sections illustrate an important principle: when you store a non-empty password using INSERT or UPDATE statements, you must use the PASSWORD()

function to encrypt it. This is because the user table stores passwords in encrypted form, not as plaintext. If you forget that fact, you are likely to attempt to set passwords like this:

```
shell> mysql -u root mysql
mysql> INSERT INTO user (Host,User,Password)
VALUES('%','jeffrey','biscuit');
mysql> FLUSH PRIVILEGES;
```

The result is that the plaintext value 'biscuit' is stored as the password in the user table. When the user jeffrey attempts to connect to the server using this password, the mysql client encrypts it with PASSWORD() and sends the result to the server. The server compares the value in the user table (which is the plaintext value 'biscuit') to the encrypted password (which is *not* 'biscuit'). The comparison fails and the server rejects the connection:

```
shell> mysql -u jeffrey -pbiscuit test
Access denied
```

Since passwords must be encrypted when they are inserted in the user table, the INSERT statement should have been specified like this instead:

```
mysql> INSERT INTO user (Host,User,Password)
VALUES('%','jeffrey',PASSWORD('biscuit'));
```

You must also use the PASSWORD() function when you use SET PASSWORD statements:

```
mysql> SET PASSWORD FOR jeffrey@%" = PASSWORD('biscuit');
```

If you set passwords using the GRANT ... IDENTIFIED BY statement or the mysqladmin password command, the PASSWORD() function is unnecessary. They both take care of encrypting the password for you, so you would specify a password of 'biscuit' like this:

```
mysql> GRANT USAGE ON *.* TO jeffrey@%" IDENTIFIED BY 'biscuit';
```

or

```
shell> mysqladmin -u jeffrey password biscuit
```

*Note: PASSWORD() does not perform password encryption in the same way that Unix passwords are encrypted. You should not assume that if your Unix password and your MySQL password are the same, PASSWORD() will result in the same encrypted value as is stored in the Unix password file. See Section 6.4 [User names], page 108.

6.15 Causes of Access denied errors

If you encounter Access denied errors when you try to connect to the MySQL server, the list below indicates some courses of action you can take to correct the problem:

- Did you run the mysql_install_db script after installing MySQL, to set up the initial grant table contents? If not, do so. See Section 6.12 [Default privileges], page 121. Test the initial privileges by executing this command:

```
shell> mysql -u root test
```

The server should let you connect without error. You should also make sure you have a file 'user.MYD' in the MySQL database directory. Ordinarily, this is 'PATH/var/mysql/user.MYD', where PATH is the pathname to the MySQL installation root.

7 MySQL language reference

7.1 Literals: how to write strings and numbers

7.1.1 Strings

A string is a sequence of characters, surrounded by either single quote (‘’’) or double quote (“”) characters (the later only if you don’t run in ANSI mode). Examples:

```
'a string'
"another string"
```

Within a string, certain sequences have special meaning. Each of these sequences begins with a backslash (‘\’), known as the *escape character*. MySQL recognizes the following escape sequences:

\0	An ASCII 0 (NUL) character.
\n	A newline character.
\t	A tab character.
\r	A carriage return character.
\b	A backspace character.
\'	A single quote (‘’’) character.
\"	A double quote (“”) character.
\\	A backslash (‘\’) character.
\%	A ‘%’ character. This is used to search for literal instances of ‘%’ in contexts where ‘%’ would otherwise be interpreted as a wildcard character.
_	A ‘_’ character. This is used to search for literal instances of ‘_’ in contexts where ‘_’ would otherwise be interpreted as a wildcard character.

Note that if you use ‘\%’ or ‘_’ in some string contexts, these will return the strings ‘\%’ and ‘_’ and not ‘%’ and ‘_’.

There are several ways to include quotes within a string:

- A ‘’’ inside a string quoted with ‘’’ may be written as ‘’’’.
- A “” inside a string quoted with “” may be written as “””.
- You can precede the quote character with an escape character (‘\’).
- A ‘’’ inside a string quoted with “” needs no special treatment and need not be doubled or escaped. In the same way, “” inside a string quoted with ‘’’ needs no special treatment.

USE_MB	This character set has multi-byte characters, handled by <code>ismbhead()</code> and <code>mbcharlen()</code>
USE_MB_IDENT	(optional) If defined, you can use table and column names that use multi-byte characters
<code>ismbchar(p, e)</code>	return 0 if <code>p</code> is not a multi-byte character string, or the size of the character (in bytes) if it is. <code>p</code> and <code>e</code> point to the beginning and end of the string. Check from <code>(char*)p</code> to <code>(char*)e-1</code> .
<code>ismbhead(c)</code>	True if <code>c</code> is the first character of a multi-byte character string
<code>mbcharlen(c)</code>	Size of a multi-byte character string if <code>c</code> is the first character of such a string
MBMAXLEN	Size in bytes of the largest character in the set

9.2 The update log

When started with the `--log-update=file_name` option, `mysqld` writes a log file containing all SQL commands that update data. The file is written in the data directory and has a name of `file_name.#`, where `#` is a number that is incremented each time you execute `mysqladmin refresh` or `mysqladmin flush-logs`, the `FLUSH LOGS` statement, or restart the server.

If you use the `--log` or `-l` options, `mysqld` writes a general log with a filename of `'hostname.log'`, and restarts and refreshes do not cause a new log file to be generated (although it is closed and reopened). By default, the `mysql.server` script starts the MySQL server with the `-l` option. If you need better performance when you start using MySQL in a production environment, you can remove the `-l` option from `mysql.server`.

Update logging is smart since it logs only statements that really update data. So an `UPDATE` or a `DELETE` with a `WHERE` that finds no rows is not written to the log. It even skips `UPDATE` statements that set a column to the value it already has.

If you want to update a database from update log files, you could do the following (assuming your update logs have names of the form `'file_name.#'`):

```
shell> ls -l -t -r file_name.[0-9]* | xargs cat | mysql
```

`ls` is used to get all the log files in the right order.

This can be useful if you have to revert to backup files after a crash and you want to redo the updates that occurred between the time of the backup and the crash.

You can also use the update logs when you have a mirrored database on another host and you want to replicate the changes that have been made to the master database.

9.3 How big MySQL tables can be

MySQL 3.22 has a 4G limit on table size. With the new MyISAM in MySQL 3.23 the maximum table size is pushed up to 8 million terabytes (2^{63} bytes).

Note however that operating systems have their own file size limits. On Linux, the current limit is 2G; on Solaris 2.5.1, the limit is 4G; on Solaris 2.6, the limit is 1000G. This means that the table size for **MySQL** is normally limited by the operating system.

By default, **MySQL** tables have a maximum size of about 4G. You can check the maximum table size for a table with the `SHOW TABLE STATUS` command or with the `myisamchk -dv table_name`. See Section 7.21 [SHOW], page 209.

If you need bigger tables than 4G (and your operating system supports this), you should set the `AVG_ROW_LENGTH` and `MAX_ROWS` parameter when you create your table. See Section 7.7 [CREATE TABLE], page 185. You can also set these later with `ALTER TABLE`. See Section 7.8 [ALTER TABLE], page 191.

If your big table is going to be read-only, you could use `myisampack` to merge and compress many tables to one. `myisampack` usually compresses a table by at least 50%, so you can have, in effect, much bigger tables. See Section 12.5 [myisampack], page 307.

Another solution can be the included `MERGE` library, which allows you to handle a collection of identical tables as one. (Identical in this case means that all tables are created with identical column information.) Currently `MERGE` can only be used to scan a collection of tables because it doesn't support indexes. We will add indexes to this in the near future.

9.4 MySQL table types

With **MySQL** you can currently (version 3.23.6) choose between three basic table formats. When you create a new table, you can tell **MySQL** which table type it should use for the table. **MySQL** will always create a `.frm` file to hold the table and column definitions. Depending on the table type the index and data will be stored in other files.

You can convert tables between different types with the `ALTER TABLE` statement. See Section 7.8 [ALTER TABLE], page 191.

•MyISAM

MyISAM is the default table type in **MySQL** 3.23. It's based on the **ISAM** code and has a lot of useful extensions.

The index is stored in a file with the `.MYI` (**MY**index) extension and the data is stored in file with the `.MYD` (**MY**Data) extension. You can check/repair **MyISAM** tables with the `myisamchk` utility. See Section 13.4 [Crash recovery], page 324.

The following is new in **MyISAM**:

- One can `INSERT` new rows in a table without deleted rows, at the same times as other threads are reading from the table.
- Supports for big files (63 bit) on filesystems/operating systems that support big files.
- All data are stored with low byte first. This makes the data machine and OS independent. The only requirement is that the machine uses 2 complement signed integers (as every machine for the last 20 years has) and IEEE floating point format (also totally dominant among mainstream machines).

PHP3 Manual

Stig Sæther Bakken

Alexander Aulbach

Egon Schmid

Jim Winstead

Lars Torben Wilson

Rasmus Lerdorf

Zeev Suraski

Edited by

Stig Sæther Bakken



Chapter 1. Introduction

What is PHP?

PHP is a server-side HTML-embedded scripting language.

Simple answer, but what does that mean? An example:

Example 1-1. An introductory example

```
<html><head><title>Example</title>
<body>
<?php echo "Hi, I'm a PHP script!"; ?>
</body></html>
```

Notice how this is different from a CGI script written in other languages like Perl or C – instead of writing a program with lots of commands to output HTML, you write an HTML script with a some embedded code to do something (in this case, output some text). The PHP code is enclosed in special start and end tags that allow you to jump into and out of "PHP mode".

What distinguishes PHP from something like client-side Javascript is that the code is executed on the server. If you were to have a script similar to the above on your server, the client would receive the results of running that script, with no way of determining what the underlying code may be. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

What can PHP do?

At the most basic level, PHP can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies.

Perhaps the strongest and most significant feature in PHP is its support for a wide range of databases. Writing a database-enabled web page is incredibly simple. The following databases are currently supported:

Adabas D	InterBase	Solid
dBase	mSQL	Sybase
Empress	MySQL	Velocis

Chapter 2. Installation

Downloading the latest version

The source code, and binary distributions for some platforms (including Windows), can be found at <http://www.php.net/>.

Installation on UNIX systems

This section will guide you through the configuration and installation of PHP. Prerequisite knowledge and software:

- Basic UNIX skills (being able to operate "make" and a C compiler)
- An ANSI C compiler
- A web server

Quick Installation Instructions (Apache Module Version)

```
1. gunzip apache_1.3.x.tar.gz
2. tar xvf apache_1.3.x.tar
3. gunzip php-3.0.x.tar.gz
4. tar xvf php-3.0.x.tar
5. cd apache_1.3.x
6. ./configure -prefix=/www
7. cd ../php-3.0.x
8. ./configure -with-mysql -with-apache=../apache_1.3.x -enable-track-vars
9. make
10. make install
11. cd ../apache_1.3.x
12. ./configure -prefix=/www -activate-module=src/modules/php3/libphp3.a
13. make
14. make install
```

Instead of this step you may prefer to simply copy the httpd binary overtop of your existing binary. Make sure you shut down your server first though.

Chapter 3. Configuration

The php3.ini file

The `php3.ini` file is read when PHP's parser starts up. For the server module versions of PHP, this happens only once when the web server is started. For the CGI version, it happens on every invocation.

Just about every directive listed here has a corresponding Apache `httpd.conf` directive. Simply prepend `php3_` to the start of the directive names listed here.

You can view the settings of most of the configuration values in the output of `phpinfo`.

General Configuration Directives

auto_append_file string

Specifies the name of a file that is automatically parsed after the main file. The file is included as if it was called with the `include` function, so `include_path` is used.

The special value `none` disables auto-appending.

Note: If the script is terminated with `exit`, auto-append will *not* occur.

auto_prepend_file string

Specifies the name of a file that is automatically parsed before the main file. The file is included as if it was called with the `include` function, so `include_path` is used.

The special value `none` disables auto-prepending.

cgi_ext string

display_errors boolean

This determines whether errors should be printed to the screen as part of the HTML output or not.

Chapter 4. Basic syntax

Escaping from HTML

There are four ways of escaping from HTML and entering "PHP code mode":

Example 4-1. Ways of escaping from HTML

1. `<? echo ("this is the simplest, an SGML processing instruction\n"); ?>`
2. `<?php echo("if you want to serve XML documents, do like this\n"); ?>`
3. `<script language="php">
 echo ("some editors (like FrontPage) don't
 like processing instructions");
</script>`
4. `<% echo ("You may optionally use ASP-style tags"); %>
 <%= $variable; # This is a shortcut for "<echo .." %>`

The first way is only available if short tags have been enabled (either by calling `short_tags`, they are configured on using the `short_tags` run-time configuration setting, or they are enabled using the `-enable-short-tags` compile-time configuration setting.

The fourth way is only available if ASP-style tags have been enabled using either the `asp_tags` configuration setting or the `-enable-asp-tags` compile-time configuration setting.

Note: Support for ASP-style tags was added in 3.0.4.

The closing "bracket" for the block will include the immediately trailing newline if one is present.

Instruction separation

Instructions are separated the same as in C or perl - terminate each statement with a semicolon.

The closing tag (`?>`) also implies the end of the statement, so the following are equivalent:

```
<?php  
    echo "This is a test";  
?>
```


Chapter 5. Types

PHP supports the following types:

- integer
- floating-point numbers
- string
- array
- object

The type of a variable is usually not set by the programmer; rather, it is decided at runtime by PHP depending on the context in which that variable is used.

If you would like to force a variable to be converted to a certain type, you may either cast the variable or use the `settype` function on it.

Note that a variable may behave in different manners in certain situations, depending on what type it is at the time. For more information, see the section on Type Juggling.

Integers

Integers can be specified using any of the following syntaxes:

```
$a = 1234; # decimal number
$a = -123; # a negative number
$a = 0123; # octal number (equivalent to 83 decimal)
$a = 0x12; # hexadecimal number (equivalent to 18 decimal)
```

Floating point numbers

Floating point numbers ("doubles") can be specified using any of the following syntaxes:

```
$a = 1.234;
$a = 1.2e3;
```

Strings

Strings can be specified using one of two sets of delimiters.

Chapter 6. Variables

Variable scope

The scope of a variable is the context within which it is defined. For the most part all PHP variables only have a single scope. However, within user-defined functions a local function scope is introduced. Any variable used inside a function is by default limited to the local function scope. For example: ...

```
$a = 1; /* global scope */

Function Test () {
    echo $a; /* reference to local scope variable */
}

Test ();
```

This script will not produce any output because the echo statement refers to a local version of the \$a variable, and it has not been assigned a value within this scope. You may notice that this is a little bit different from the C language in that global variables in C are automatically available to functions unless specifically overridden by a local definition. This can cause some problems in that people may inadvertently change a global variable. In PHP global variables must be declared global inside a function if they are going to be used in that function. An example:

```
$a = 1;
$b = 2;

Function Sum () {
    global $a, $b;

    $b = $a + $b;
}

Sum ();
echo $b;
```

The above script will output "3". By declaring \$a and \$b global within the function, all references to either variable will refer to the global version. There is no limit to the number of global variables that can be manipulated by a function.

Chapter 7. Constants

PHP defines several constants and provides a mechanism for defining more at run-time. Constants are much like variables, save for the two facts that constants must be defined using the `define` function, and that they cannot later be redefined to another value.

The predefined constants (always available) are:

`__FILE__`

The name of the script file presently being parsed. If used within a file which has been included or required, then the name of the included file is given, and not the name of the parent file.

`__LINE__`

The number of the line within the current script file which is being parsed. If used within a file which has been included or required, then the position within the included file is given.

`PHP_VERSION`

The string representation of the version of the PHP parser presently in use; e.g. '3.0.8-dev'.

`PHP_OS`

The name of the operating system on which the PHP parser is executing; e.g. 'Linux'.

`TRUE`

A true value.

`FALSE`

A false value.

`E_ERROR`

Denotes an error other than a parsing error from which recovery is not possible.

`E_WARNING`

Denotes a condition where PHP knows something is wrong, but will continue anyway; these can be caught by the script itself. An example would be an invalid regexp in `ereg`.

Chapter 8. Expressions

Expressions are the most important building stones of PHP. In PHP, almost anything you write is an expression. The simplest yet most accurate way to define an expression is "anything that has a value".

The most basic forms of expressions are constants and variables. When you type "`$a = 5`", you're assigning '5' into `$a`. '5', obviously, has the value 5, or in other words '5' is an expression with the value of 5 (in this case, '5' is an integer constant).

After this assignment, you'd expect `$a`'s value to be 5 as well, so if you wrote `$b = $a`, you'd expect it to behave just as if you wrote `$b = 5`. In other words, `$a` is an expression with the value of 5 as well. If everything works right, this is exactly what will happen.

Slightly more complex examples for expressions are functions. For instance, consider the following function:

```
function foo () {  
    return 5;  
}
```

Assuming you're familiar with the concept of functions (if you're not, take a look at the chapter about functions), you'd assume that typing `$c = foo()` is essentially just like writing `$c = 5`, and you're right. Functions are expressions with the value of their return value. Since `foo()` returns 5, the value of the expression '`foo()`' is 5. Usually functions don't just return a static value but compute something.

Of course, values in PHP don't have to be integers, and very often they aren't. PHP supports three scalar value types: integer values, floating point values and string values (scalar values are values that you can't 'break' into smaller pieces, unlike arrays, for instance). PHP also supports two composite (non-scalar) types: arrays and objects. Each of these value types can be assigned into variables or returned from functions.

So far, users of PHP/FI 2 shouldn't feel any change. However, PHP takes expressions much further, in the same way many other languages do. PHP is an expression-oriented language, in the sense that almost everything is an expression. Consider the example we've already dealt with, '`$a = 5`'. It's easy to see that there are two values involved here, the value of the integer constant '5', and the value of `$a` which is being updated to 5 as well. But the truth is that there's one additional value involved here, and that's the value of the assignment itself. The assignment itself evaluates to the assigned value, in this case 5. In practice, it means that '`$a = 5`', regardless of what it does, is an expression with the value 5. Thus, writing something like '`$b = ($a = 5)`' is like writing '`$a = 5; $b = 5;`' (a semicolon marks the end of a statement). Since assignments are parsed in a right to left order, you can also write '`$b = $a = 5`'.

Another good example of expression orientation is pre- and post-increment and decrement. Users of PHP/FI 2 and many other languages may be familiar with the notation of `variable++` and `variable--`. These are increment and decrement operators. In PHP/FI 2, the statement '`$a++`' has no value (is not an

Chapter 9. Operators

Arithmetic Operators

Remember basic arithmetic from school? These work just like those.

Table 9-1. Arithmetic Operators

example	name	result
$\$a + \b	Addition	Sum of $\$a$ and $\$b$.
$\$a - \b	Subtraction	Remainder of $\$b$ subtracted from $\$a$.
$\$a * \b	Multiplication	Product of $\$a$ and $\$b$.
$\$a / \b	Division	Dividend of $\$a$ and $\$b$.
$\$a \% \b	Modulus	Remainder of $\$a$ divided by $\$b$.

String Operators

There is only really one string operator – the concatenation operator (".").

```
$a = "Hello ";  
$b = $a . "World!"; // now $b = "Hello World!"
```

Assignment Operators

The basic assignment operator is "=". Your first inclination might be to think of this as "equal to". Don't. It really means that the the left operand gets set to the value of the expression on the rights (that is, "gets set to").

The value of an assignment expression is the value assigned. That is, the value of " $\$a = 3$ " is 3. This allows you to do some tricky things:

Chapter 10. Control Structures

Any PHP script is built out of a series of statements. A statement can be an assignment, a function call, a loop, a conditional statement or even a statement that does nothing (an empty statement). Statements usually end with a semicolon. In addition, statements can be grouped into a statement-group by encapsulating a group of statements with curly braces. A statement-group is a statement by itself as well. The various statement types are described in this chapter.

if

The `if` construct is one of the most important features of many languages, PHP included. It allows for conditional execution of code fragments. PHP features an `if` structure that is similar to that of C:

```
if (expr)
    statement
```

As described in the section about expressions, `expr` is evaluated to its truth value. If `expr` evaluates to `TRUE`, PHP will execute `statement`, and if it evaluates to `FALSE` - it'll ignore it.

The following example would display `a is bigger than b` if `$a` is bigger than `$b`:

```
if ($a > $b)
    print "a is bigger than b";
```

Often you'd want to have more than one statement to be executed conditionally. Of course, there's no need to wrap each statement with an `if` clause. Instead, you can group several statements into a statement group. For example, this code would display `a is bigger than b` if `$a` is bigger than `$b`, and would then assign the value of `$a` into `$b`:

```
if ($a > $b) {
    print "a is bigger than b";
    $b = $a;
}
```

If statements can be nested indefinitely within other `if` statements, which provides you with complete flexibility for conditional execution of the various parts of your program.

else