

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
Facultad de Contaduría y Administración
Facultad de Ciencias Químicas e Ingeniería
Maestría en Tecnologías de la Información y la Comunicación



**Diseño e implementación de Software para la
administración del centro de cómputo de la
Facultad de Contaduría y Administración.**

**TESIS QUE PARA OBTENER EL GRADO DE
MAESTRA EN TECNOLOGIAS DE LA INFORMACIÓN Y LA COMUNICACIÓN**

Presenta:

I.S.C. Anneliesse Margarita Crabtree García

Bajo la dirección de:

M.C. Cesar Ortega Corral

Codirección:

M.G.A. María Elizabeth Ojeda
Orta

TIJUANA, BAJA CALIFORNIA, MÉXICO

ABRIL 2013

Mi dedicatoria

Dedico este trabajo a las dos personas más importantes en mi vida que son mis padres por su gran amor, paciencia y apoyo incondicional, que sin ellos no hubiera sido posible obtener este nuevo éxito en nuestra vida porque de los tres los amo. Así mismo quiero agradecer a mi padre dios por bendecirme que estén mis padres a mi lado así como otorgarme la salud, y todo lo necesario para culminar esta nueva meta en mi vida.

Agradecimientos

Deseo agradecer a esta gran institución que es la Universidad Autónoma de Baja California así como M.C. Julio Octavio Blas Flores que el siendo director de la Facultas de Contaduría y Administración, en el tiempo de estudio me otorgaron la beca económica y el apoyo que necesario para poder realizar esta meta y sueño que tenía desde tiempo atrás.

A MIS AMIGAS.

Un agradecimiento y reconocimiento muy especial quiero hacer a mis dos grandes amigas la Mtra. Karina Raya Díaz y la Mtra. Elizabeth Ojeda Orta que sin amistad apoyo y empuje para los momentos de flaquees, hubiera logrado obtener este producto, Karina con sus ánimos, alegría y búsqueda de información que debía leer, Elizabeth con sus grandes conocimientos de metodóloga guiándome en la redacción y claro siendo estricta en los momentos apropiados. Gracias...

A MIS HERMANOS.

Quiero agradecer a mi hermana Monica por su gran apoyo en momentos difíciles a su gran optimismo y ejemplo para conseguir todas las metas que uno se propone, a Josefina por su gran ternura, preocupación y cariño. Carlos (Chale) por su ánimos.

A MIS COMPAÑEROS MAESTROS Y AMIGOS.

Por sus ánimos, alegrías, tristezas y tantos momentos compartidos con todos ustedes. No quiero cerrar estos agradecimientos sin antes dárselo a la Dra. Margarita Ramírez que como coordinadora de la MTIC y a siempre ha estado dándonos ánimo, acariño y tiempo. Así como a mi director de Tesis M.C. Cesar Ortega Corral por tu tiempo para leer y realizar las correcciones y observaciones necesarias a la Dra. Maricela Sevilla Caro por sus observaciones.

Anneliesse Margarita Crabtree Garcia.

Resumen.

En este documento se describe el proceso que se lleva a cabo en el desarrollo de software para la administración del centro de cómputo que entre otros, cuenta con los servicios de préstamo de laboratorios para la impartición de clases. Uno de los beneficios de este software es mejorar el control de acceso a los laboratorios que son para el uso de docentes, estudiantes y personal administrativo.

En la propuesta e implementación del sistema se incluye la generación de reportes para el análisis de espacios para la distribución de clases, logrando con ello un aprovechamiento de las instalaciones; así como solucionar la necesidad de poder tener información que le ayude a tomar decisiones en cuestión de espacios y recursos físicos para un óptimo uso de tecnologías.

Índice.

Mi dedicatoria	2
Agradecimientos	4
Resumen.....	5
Índice.....	6
Capítulo I Introducción.....	8
1.1. Antecedentes de Proyecto:.....	9
1.2. Planteamiento.....	13
1.4. Objetivos.	15
1.4.1. Objetivo General:	15
1.4.2. Objetivos específicos:.....	15
1.5. Justificación.	16
Capítulo II Marco Teórico:.....	17
2.1 Administración en centros de cómputo.....	17
2.2 Software de administración de centros de cómputo.....	18
2.2.1 Evolución del Software.....	18
2.3 Sistemas de información.	18
2.3.1 Sistemas de información estratégica.	18
2.3.3 Sistemas de administración de base de datos.	19
2.3.4 Sistemas operativos.	19
2.3.5 Sistemas de apoyo a la toma de decisiones.....	19
2.4 Redes y Telecomunicaciones.	20
2.4.1 La comunicación es la clave.	21
2.4.2 Redes.	21
2.4.3 Protocolos.	22
2.4.4 Internet.....	22
2.5 Base de datos y almacenes de datos.....	22
2.5.1 Bases de datos.....	23
2.5.2 Bases de datos en la WEB.	23
2.5.3 Administración de bases de datos	23
2.7 Inteligencia de negocios y administración.	25
2.7.1 Software de gestión de proyecto.	25
2.7.2 Procesamiento analítico en línea.	25
2.8 Lenguaje Unificado de Modelado	25
2.8.1 Diagrama de Casos de Uso	26

2.10 PHP& MySQL como herramienta de desarrollo	27
2.10.1 Breve Historia de PHP	27
2.10.2 Lenguaje de programación interpretado PHP.....	28
Capítulo III Metodología.....	29
3.1. Modelo de la Cascada.	29
3.1.1. Comunicación.....	47
3.1.2. Planeación.	47
3.1.2.1. Estudio de factibilidad:.....	47
3.1.3. Modelado.	48
3.1.3.1. Análisis.....	49
3.1.3.2. Diseño.....	49
3.1.4. Construcción.....	50
3.1.4.1. Prueba	50
3.1.5. Despliegue.....	51
3.1.5.1. Mantenimiento	51
3.2. Sistema	51
Capítulo IV. Desarrollo	52
4.1. Arquitectura de Software.....	52
4.2. Diseño de base de datos	53
4.3. Casos de Uso.	54
4.4. Software de gestión	60
4.4.1. Descripción de Software de gestión de laboratorios.....	60
4.4.1.1. Inicio de Sección del usuario:	60
4.4.1.2. Módulo de reservación de salones para el docente:	61
4.4.1.2.1. Eliminación de reservaciones.....	64
4.4.1.2.2. Consultar reservaciones.....	66
4.4.1.3. <i>Módulo de grupos:</i>	67
4.4.1.4. <i>Módulo de salones.</i>	69
4.4.1.5. <i>Información personal:</i>	70
4.4.1.6. <i>Menú principal para el administrador.</i>	70
4.5. Pruebas estáticas.....	74
Capítulo V Resultados.	75
Capítulo VI Conclusiones	82
Conclusiones.....	82
Índice de Siglas	84
Bibliografía	84
Cibergrafía	84

Capítulo I Introducción.

Un centro de cómputo representa una parte esencial dentro de toda organización, esto incluye a las entidades educativas, las cuales en los últimos años se han visto obligadas a modificar sus sistemas organizacionales, estructurales y culturales para alcanzar los niveles de desempeño requeridos debido a la globalización, lo que se ha traducido en fuertes inversiones de recursos en cuanto a infraestructura se refiere. De ahí la importancia de tener una adecuada administración de los recursos informáticos, la cual sirve de apoyo al docente para el ejercicio de su profesión académica, y al alumno para que cuente con las mejores condiciones para realizar su trabajo. Aunado a lo anterior, los centros de cómputo de calidad aspiran bajo el escrutinio de organismos acreditadores nacionales, los cuales se encargan de vigilar que en las Instituciones de Educación Superior se cumplan con una serie de mecanismos y formas, mediante las cuales se obtiene la comprobación de que sus procesos se desempeñen con determinados estándares de calidad académica para impartir estudios a nivel superior. Y es principalmente para cumplir con ciertos criterios establecidos por estos organismos, que surge la necesidad de la mejora de dichos procesos, aplicando la innovación tecnológica, a través del desarrollo e implementación del software para la gestión del centro de cómputo.

En el presente documento se plantea la problemática que justifica el desarrollo de este software, así como el planteamiento de los objetivos que se pretenden lograr en este trabajo.

1.1. Antecedentes de Proyecto:

La tecnología de la informática y las comunicaciones han modificado de manera importante la forma de trabajar de toda la humanidad. La dependencia que tenemos en la actualidad de las computadoras es evidente; sin embargo, los sucesos han ocurrido con tal rapidez que toda empresa depende de la tecnología para hacer cualquier operación, y de esto nos damos cuenta que el desconectar todas las máquinas ocasionarían mas desastre que una guerra en la humanidad.

No se puede evaluar realmente el grado de dependencia de la tecnología pero es innegable la frecuencia con que las personas son víctimas de errores que se cometen en centros de cómputo. ¿Quién no ha perdido horas esperando a que se restablezcan los servicios de algún banco para cobrar un cheque? ¿Cuántas veces no hemos acudido a aclarar un recibo de cobro emitido erróneamente? Miles de trámites han quedado pendientes por falta de información oportuna.

La función de la informática es simplificar las labores administrativas, estas en ciertos casos terminan por ser más complicadas. (Jiménez, 2006) El problema que se tiene en los centros de cómputo radica en un bajo nivel administrativo de las áreas de informática, y por lo tanto es importante el tener una área de informática bien administrada. En las instituciones educativas no quedan exentas de tener una escasa administración así se constata en el centro de cómputo de la Universidad Autónoma de Baja California (UABC) campus Tijuana, en la Facultad de Contaduría y Administración (FCA), que para poder ofrecer el servicio a más de 4000 alumnos por semestres distribuidos en las Licenciaturas de Informática, Contabilidad, Administración de Empresas y Negocios internacionales; además de las maestrías en Tecnologías de la Información, y Contabilidad, Doctorado en Administración. Debido a que en sus planes de estudio incorporan tecnologías de información y comunicación para estar a la vanguardia cumpliendo con los objetivos de poder brindar a la comunidad egresados con un alto grado de calidad.

La FCA cuenta con la infraestructura física que se en lista a continuación:

- Tres edificios con 16 de salones para impartir clases.
- Un edificio de emprendedores.

- Un Edificio para dirección que incluye oficinas administrativas así como los cubículos de los docentes de posgrado.
- Una sala de juntas.
- 2 salas de uso común (Conferencias, 1 Audiovisual con capacidad de 150 personas, 1 salón de conferencia para 50 personas).
- Un centro de cómputo con 10 laboratorios de cómputo de los cuales 7 de ellos cuentan con 36 equipos de cómputo, 9 de ellos con proyector y pizarrón electrónico, 2 laboratorios más con 20 equipos y uno más equipado para la siguientes clases: Redes, arquitectura de computadoras entre otras, con el siguiente mobiliario y equipo: mesas, monitores y herramienta para la impartición de las clases antes mencionadas.
- Un centro de cómputo
 - 7 laboratorios de cómputo con proyector y 36 equipos.
 - 2 laboratorios de cómputo con proyector y 20 equipos.
 - 1 laboratorio de redes y arquitectura de computadoras.
 - Dispositivos de red: *rack, patch panels, switch, routers*, etc.
 - Mesas de trabajo.
 - Almacén de herramientas.

Para ver cómo está representada la estructura organizacional de la FCA se integra el organigrama en el cual el centro de cómputo da servicio a todas las áreas que la integran en cuestiones tecnológicas.

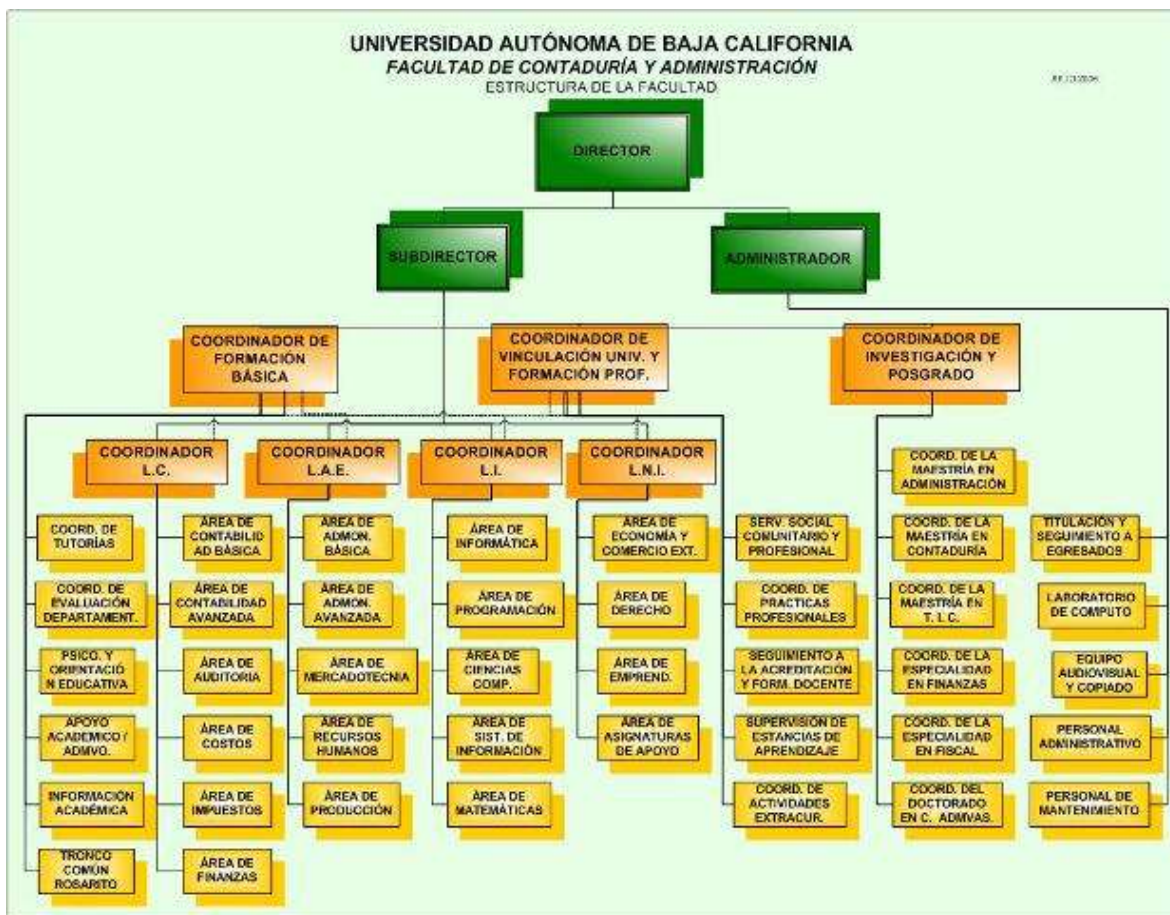


Figura 1. Estructura organizacional de la FCA.

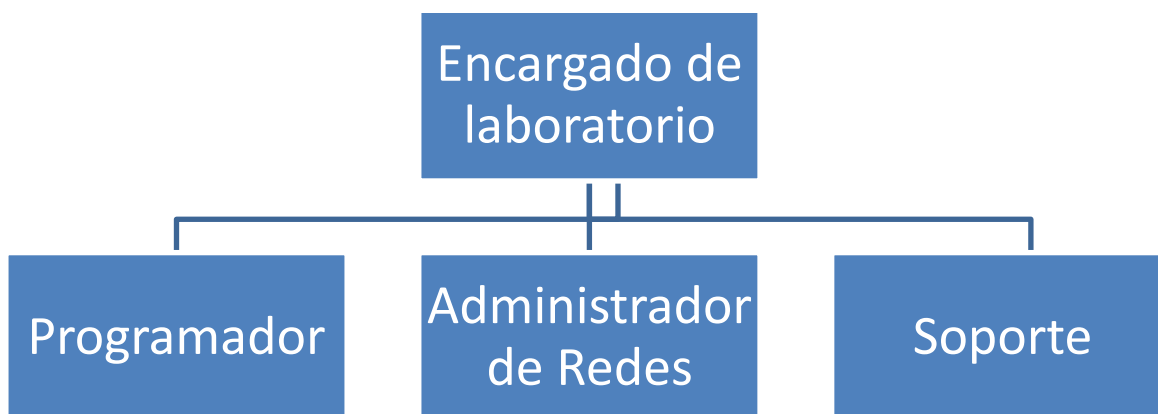


Figura 2. Estructura organizacional del centro de cómputo.

Una actividad importante para la impartición de clases es la reservación del uso de laboratorios de la FCA, esto lo realiza el administrador del laboratorio de la siguiente manera:

1. Solicita los horarios de maestros, con las materias a impartir por los maestros al subdirector académico antes de inicio de clases.
2. Revisar horas laboratorio de las cartas descriptivas las cuales describen el contenido temático, las horas clase y de laboratorio por materia.
3. Elaborar calendario ver figura 3.
4. Los maestros que imparten materias sin horas laboratorio asignadas en las cartas descriptivas, y requieren, las solicitarán administrador del laboratorio para que éste se las asigne.
5. Para reservar un salón se tiene que revisar que el salón, tenga los programas que requiera la materia, el número de máquinas, proyector, etc.
6. Los datos anotados en los horarios son, numero de grupo, nombre del maestro que imparte la clase y clave de la materia, los anteriores datos son capturados en las celdas donde corresponda el día y la hora en caso que sea más de una hora se repiten los datos en todas las horas que se va ocupar.
7. Los casos en que es un solo día el evento o no más de una semana se apunta con lápiz y se agrega la fecha para
8. su posterior borrado de la carpeta no se captura en el archivo electrónico y por lo tanto no se imprimir hojas para sustituir las hojas pegadas en los salones.

	DOMINGO	LUNES	MARTES	MIERCOLES	VIERNES
07:00 - 08:00	802 CARINA GARCIA FAVELA 11242	802 MICHON ENRIQUE 8612	802 CARINA GARCIA FAVELA 11242		802 VANESSA FABIOLA GARRIDO 11242
08:00 - 09:00	802 CARINA GARCIA FAVELA 11242	804 MARIA ELIZABETH ZEPEDA 11242	802 CARINA GARCIA FAVELA 11242	804 MARIA ELIZABETH ZEPEDA 11242	802 VANESSA FABIOLA GARRIDO 11242
09:00 - 10:00	801 ESPERANZA MARRIQUE ROJAS 11242	804 MARIA ELIZABETH ZEPEDA 11242	801 MARIA ELIZABETH ZEPEDA 11242	804 MARIA ELIZABETH ZEPEDA 11242	
10:00 - 11:00	801 ESPERANZA MARRIQUE ROJAS 11242	801 EDUARDO ABRAMA TELLO 802	801 MARIA ELIZABETH ZEPEDA 11242	802 COCOS 11242 802C	801 MARIA ELIZABETH ZEPEDA 11242
11:00 - 12:00	801 ESPERANZA MARRIQUE ROJAS 8612	801 EDUARDO ABRAMA TELLO 802	802 CORONADO ABRAMA TELLO 8612	801 ESPERANZA MARRIQUE ROJAS 8612	801 MARIA ELIZABETH ZEPEDA 11242
12:00 - 13:00	801 ESPERANZA MARRIQUE ROJAS 8612	807 JORGE INES MORALES 11242	802 CORONADO ABRAMA TELLO 8612	801 ESPERANZA MARRIQUE ROJAS 8612	
13:00 - 14:00	802 JORGE INES MORALES 11242	807 JORGE INES MORALES 11242		802 JORGE INES MORALES 11242	807 JORGE INES MORALES 11242
14:00 - 15:00		CURSO ADULTOS NO RECONOCEN ASESOR 1A 800		CURSO ADULTOS NO RECONOCEN ASESOR 1A 800	807 JORGE INES MORALES 11242

Figura 3. Horarios de reservación de laboratorios

Fuente Elaboración propia

1.2. Planteamiento.

Como ya se mencionó las inversiones realizadas por las Instituciones de Educación Superiores (IES) en infraestructura son significativas, debido a esto la exigencia del uso más eficiente posible de las instalaciones y el equipamiento de laboratorios cada vez mayor, sobre todo en países como el nuestro donde prevalece la escasez de recursos. La administración de cualquier tipo de recurso en un centro de cómputo debe incluir la realización de las tareas propias de administración, tales como: planeación, organización, dirección y control. Sin embargo, en muchas ocasiones, no son utilizados los métodos que garanticen su adecuada solución y permitan un crecimiento equilibrado y conjunto (Jiménez, 2006).

Un centro de cómputo representa una entidad dentro de una empresa u organización, la cual tiene como objetivo satisfacer las necesidades de información de la empresa, de manera veraz y oportuna. Su función principal es apoyar la labor de la empresa mediante una más acertada toma de decisiones y así hacerla más segura fluida y simplificada [2].

De acuerdo al criterio anterior y mediante la implementación del software se pretende dar solución a los siguientes problemas:

1. Cuando no está el administrador la reservación la realiza cualquier auxiliar del laboratorio. Anotándola con pluma o lápiz en la carpeta de los horarios.
2. La carpeta está en el escritorio del administrador para consulta de maestros y auxiliares del laboratorio, pero en ocasiones se extravía.
3. La copia de los horarios clase que se pega en cada salón no siempre está actualizado al 100% y en ocasiones puede ser retirada por una persona no autorizada.
4. No hay historial de uso de salones.
5. Se asigna el salón y resulta que los equipos no tienen los programas requeridos, o los que requieren tiene conflicto con los de otras materias.
6. Cuando los técnicos o administradores están ocupados, los maestros realizan su propia reservación, sin previa autorización de horas.
7. Duplicidad de la reservación, lo cual genera conflicto entre maestros.

8. Desperdicio de recurso en tiempo de uso, por los siguientes factores:
- Maestros reservan todo el semestre horas-salón para garantizar sólo unas cuantas horas de uso al final del semestre.
 - Para utilizar la plataforma Blackboard® reservan horas para los alumnos pero no para impartir clase.
 - No hay congruencia entre las horas-laboratorio asignadas en el semestre, con respecto a las horas realmente utilizadas.
 - Dificultad para consultar la disponibilidad por parte del maestro el mismo día. Lo que le representa pérdida de tiempo clase.
 - No se reservaron las horas clase de laboratorio obligatorias por error humano.
 - Dificultad para liberar las horas no utilizadas o reservadas para que alguien más las utilice.
9. No hay información estadística de indicadores de uso para toma de decisiones de actualización de software o equipo.
10. No existe control de acceso al laboratorio de cómputo.

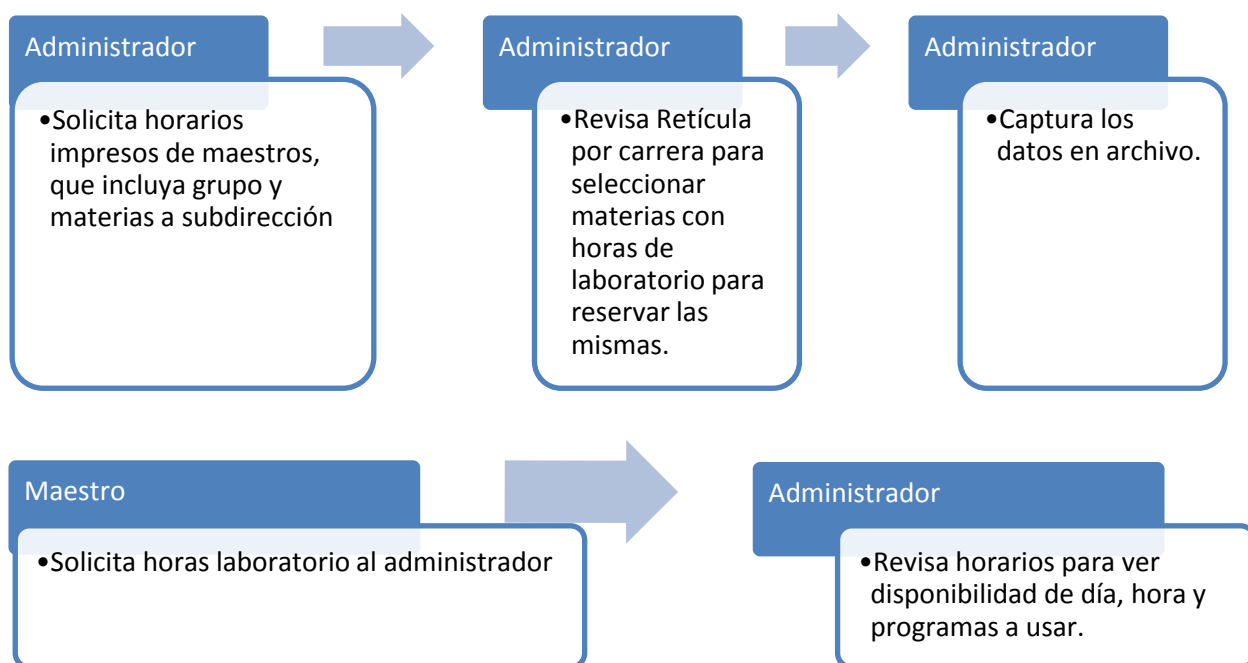


Figura 4. Diagrama de proceso actual en la reservación de laboratorios.

Fuente: Elaboración propia.

1.4. Objetivos.

1.4.1. Objetivo General:

Diseñar e implementar un software para la administración y control de los laboratorios de cómputo de la FCA de la Universidad Autónoma de Baja California.

1.4.2. Objetivos específicos:

- a) Desarrollar un software de fácil acceso desde Internet a partir del cual el personal docente podrá reservar espacios para impartir sus clases.
- b) Obtener información estadística por carrea, con respecto a la frecuencia de uso del equipo de cómputo por parte del alumnado.
- c) Desarrollar un módulo de consulta para el personal docente con respecto a los recursos de hardware y software con que cuenta cada laboratorio.
- d) Implementar estrategias que nos permitan identificar los espacios muertos de los laboratorios para promover su uso.

1.5. Justificación.

La adecuada administración de la infraestructura, la cual sirve de apoyo al docente para el ejercicio de su profesión académica y al alumno para que cuente con las mejores condiciones para realizar su trabajo, es un asunto de suma importancia. Por ello es necesario y urgente potenciar el uso del recurso informático (laboratorios y equipo) de la institución, esto se busca solucionar con un software de fácil acceso desde internet para los docentes con el cual ellos realicen las reservaciones para impartir sus clases.

El laboratorio debe tener un control del uso de equipo por parte del alumnado, profesor académico y administrativo, para detectar cualquier anomalía y prevención de daños, y adicionalmente generar información estadística para la toma de decisiones.

Capítulo II Marco Teórico:

2.1 Administración en centros de cómputo

Cuando se plantea que en el centro de cómputo de la FCA es necesario un sistema de software de administración para la operación y cumplimiento de una educación integral por parte de la facultad, nos viene a la mente que se requiere una administración adecuada para poder ofrecer el servicio que exige la institución, se tiene que iniciar por precisar qué es Administración la cual se define como el proceso de crear, diseñar y mantener un ambiente en que las personas al laborar o trabajar en grupos, alcancen con eficiencia las metas seleccionadas. (Jiménez, 2006) La administración de un centro de cómputo debe incluir la realización de las tareas propias de administración de cualquier tipo de recurso: planeación, organización, dirección y control. Sin embargo, en la mayoría de las ocasiones, no son utilizados métodos que garanticen su adecuada solución y permitan un crecimiento equilibrado y conjunto.

El término centro de cómputo a principio de los 1950's indicaba un local que alojaba equipo computacional y personal técnico que operaba tales equipos. Este concepto se ha considerado igual como centro de cómputo pero en donde los servicios que se llevan a cabo son el desarrollo y mantenimiento de sistemas informáticos, operación de sistemas de producción, operación de equipos, etc. Sin importar el tamaño de los equipos, ya que actualmente se están produciendo dispositivos más pequeños y sofisticados (Jiménez, 2006).

Un centro de cómputo representa una entidad dentro de una empresa u organización, la cual tiene como objetivo satisfacer las necesidades de información de la empresa, de manera veraz y oportuna. Su función principal es apoyar la labor de la empresa mediante una más acertada toma de decisiones y así hacerla segura, fluida y simplificada.

Dentro de estos Centros de Cómputo se encuentran, además del equipo y programas que procesan la información, los recursos humanos especializados, el factor más importante, de cuya habilidad depende la satisfacción de las necesidades de cómputo de la organización. Tiene la responsabilidad de acaparar,

centralizar, custodiar y procesar la mayoría de los datos con los que opera la compañía. Prácticamente todas las actividades de los demás departamentos se basa en la información que les procesa el centro de cómputo. La importancia que tiene el centro de cómputo dentro de la organización, lo coloca en una posición que influye incluso en una gran parte de las decisiones administrativas y de proyección de la empresa.

Su administración involucra el control tanto físico como económico y del buen funcionamiento del software y hardware, así como el desempeño del personal que labora en él. Si estos conceptos se manejan estrictamente, la fluidez de la información de entrada y salida, el mantenimiento del equipo y personal idóneo, estarán cumpliendo con las condiciones necesarias para que funcione correctamente un centro de cómputo.

2.2 Software de administración de centros de cómputo.

2.2.1 Evolución del Software.

Para poder lograr una eficiente administración se han creado una diversidad de software que ayudan con esta tarea que van desde el control de acceso a los equipos, la asignación de espacios, resguardo de equipos y software, apoyo a usuarios y las diversas actividades que se realizan en un centro de cómputo. Estos software existen tanto en forma gratuita como con un costo o también se han creado por personal que labora en estos departamentos.[1]

2.3 Sistemas de información.

Un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones (Kenneth C. Laudon, 2008).

2.3.1 Sistemas de información estratégica.

Un Sistema de Información estratégico puede ser considerado como el uso de la tecnología de la información para soportar o dar forma a la estrategia

competitiva de la organización, a su plan para incrementar o mantener la ventaja competitiva o bien reducir la ventaja de sus rivales. (OZ, 2008)

2.3.3 Sistemas de administración de base de datos.

Los sistemas de gestión de bases de datos (en inglés *database management system*, abreviado DBMS) es el software que permite a una organización centralizar los datos, administrarlos eficientemente y proporcionar, mediante los programas de aplicación, el acceso a los datos almacenados. El DBMS actúa como una interfaz entre los programas de aplicación y los archivos de datos físicos. (Kenneth C. Laudon, 2008).

2.3.4 Sistemas operativos.

Un Sistema Operativo es el software encargado de ejercer el control y coordinar el uso del hardware entre diferentes programas de aplicación y los diferentes usuarios. Es un administrador de los recursos de hardware del sistema.

En una definición informal es el software más importante del sistema, ya que se encarga de administrar todos los recursos computacionales en tareas tales como la asignación de tiempo de uso del CPU a los procesos que lo requieren, la asignación y el manejo de la memoria principal, el manejo de los dispositivos periféricos y, en general, de todos los recursos de hardware y software (Cohen Karen & Asín Lares, 2009).

2.3.5 Sistemas de apoyo a la toma de decisiones

Los sistemas de apoyo a la toma de decisiones (DSS, Decisión Support Systems) constituyen una clase de alto nivel de sistemas de información computarizada. Los DSS coinciden con los sistemas de información gerencial en que ambos dependen de una base de datos para abastecerse de datos. Sin embargo, difieren en que el DSS pone énfasis en el apoyo a la toma de decisiones en todas sus fases, aunque la decisión definitiva es responsabilidad exclusiva del encargado de tomarla. Los sistemas de apoyo a la toma de decisiones se ajustan

más al gusto de la persona o grupo que los utiliza que a los sistemas de información gerencial tradicionales. En ocasiones se hace referencia a ellos como sistemas que se enfocan en la inteligencia de negocios (Kendall, 2005).

2.4 Redes y Telecomunicaciones.

Las telecomunicaciones y las redes son parte fundamental de la “Explosión de información” de hecho son el móvil por lo cual es de suma importancia que los administradores de las empresas de hoy y del futuro entiendan.

Un sistema de telecomunicaciones consiste en una infraestructura física a través de la cual se transporta la información desde la fuente hasta el destino, y con base en esa infraestructura se ofrecen a los usuarios los diversos servicios de telecomunicaciones (figura 5). En lo sucesivo se denominará "red de telecomunicaciones" a la infraestructura encargada del transporte de la información. Para recibir un servicio de telecomunicaciones, un usuario utiliza un equipo terminal a través del cual obtiene entrada a la red por medio de un canal de acceso. Cada servicio de telecomunicaciones tiene distintas características, puede utilizar diferentes redes de transporte, y, por tanto, el usuario requiere de distintos equipos terminales. Por ejemplo, para tener acceso a la red telefónica, el equipo terminal requerido consiste en un aparato telefónico; para recibir el servicio de telefonía celular, el equipo terminal consiste en teléfonos portátiles con receptor y transmisor de radio, etcétera [9].

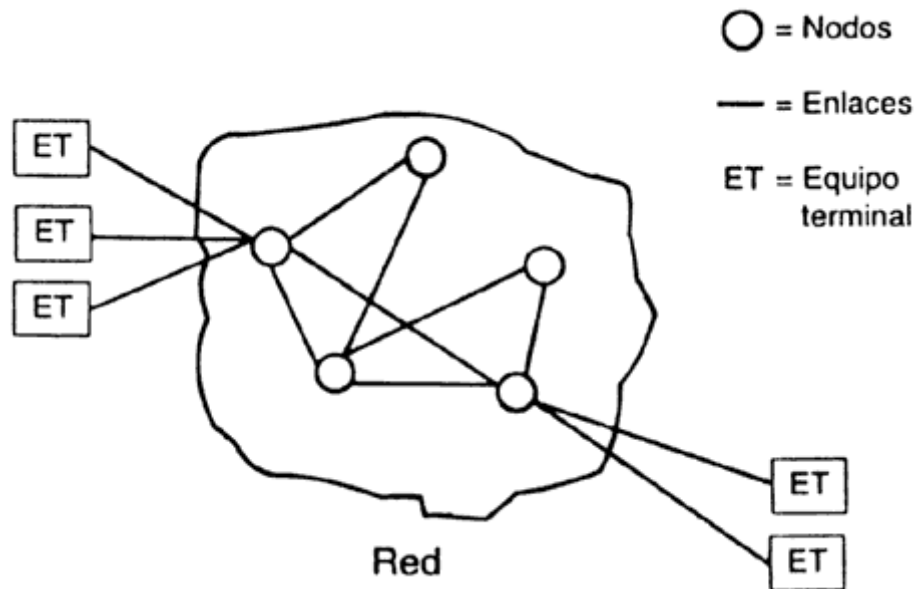


Figura 5. Diagrama de proceso actual en la reservación de laboratorios.

Fuente: [9].

2.4.1 La comunicación es la clave.

Es el traspaso de información, desde un emisor, mediante un mensaje, hacia un receptor. Proceso el cual, luego toma el sentido inverso. O sea, el receptor se vuelve emisor y así consecutivamente. Esta información, que es transmitida, por medio de la comunicación, es recibida por el subconsciente y luego captada por el consciente (Cohen Karen & Asín Lares, 2009).

2.4.2 Redes.

Las redes constan de dos o más computadoras conectadas entre sí y permiten compartir recursos e información. La información por compartir suele consistir en archivos y datos. Los recursos son los dispositivos o las áreas de almacenamiento de datos de una computadora, compartida por otra computadora mediante la red. La más simple de las redes conecta dos computadoras, permitiéndoles compartir archivos e impresos. Una red mucho más compleja conecta todas las computadoras de una empresa o compañía en el mundo. Para compartir impresoras basta con un conmutador, pero si se desea compartir

eficientemente archivos y ejecutar aplicaciones de red, hace falta tarjetas de interfaz de red (NIC, NetWare Interfaces Cards) y cables para conectar los sistemas. Aunque se puede utilizar diversos sistemas de interconexión vía los puertos series y paralelos, estos sistemas baratos no ofrecen la velocidad e integridad que necesita un sistema operativo de red seguro y con altas prestaciones que permita manejar muchos usuarios y recursos. (Parsons & Dan, 2008)

2.4.3 Protocolos.

Un protocolo es un método establecido de intercambiar datos en Internet. Un protocolo es un método por el cual dos ordenadores acuerdan comunicarse, una especificación que describe cómo los ordenadores hablan el uno al otro en una red (Parsons & Dan, 2008).

2.4.4 Internet.

Es la infraestructura de comunicación mundial que vincula redes de computadoras utilizando el protocolo TCP/IP (Parsons & Dan, 2008), cuando se logra crear esta infraestructura tan poderosa creó una revolución en casi o en todos los aspectos de la vida de la sociedad actual desde como adquirir algún producto hasta aspectos sociales de como interactuar o conocer persona y también al momento de tener el internet disponible la instituciones educativas empezaron a cambiar su forma de impartir la educación y sus procesos administrativos.

2.5 Base de datos y almacenes de datos.

En la actualidad la información es considerada el activo más importante de las empresas después del humano, esa riqueza de la información se refiere a la complejidad, distribución, almacenamiento y forma de acceso a la misma.

Un factor importante es el almacenamiento de toda la información que generan las entidades por medio de sus sistemas y de aquí la importancia de

seleccionar la tecnología adecuada. Para el sistema que se presenta en este documento se seleccionó el uso de “Bases de Datos” y las razones fueron que:

1. Proporciona a los usuarios el acceso a datos, que pueden visualizar, ingresar o actualizar, en concordancia con los derechos de acceso que se les hayan otorgado. Se convierte más útil a medida que la cantidad de datos almacenados crece.
2. Una base de datos puede ser local, es decir que puede utilizarla sólo un usuario en un equipo, o puede ser distribuida, es decir que la información se almacena en equipos remotos y se puede acceder a ella a través de una red.
3. La principal ventaja de utilizar bases de datos es que múltiples usuarios pueden acceder a ellas al mismo tiempo [15].

2.5.1 Bases de datos.

Es importante el almacenamiento y control de la información por lo consiguiente es estratégico definir el medio de almacenamiento de cualquier sistema, revisando la definición de lo que es una base de datos o banco de datos que es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso [10] se determina que esta va ser la forma de almacenar los datos del sistema a desarrollar.

2.5.2 Bases de datos en la WEB.

Base de Datos Web es una herramienta que organiza y administra la información de forma sencilla a través de una interfaz Web [12].

2.5.3 Administración de bases de datos

Con el surgimiento de poder controlar y tener la seguridad que la información este segura y bien gestionada surge la necesidad de contar con un sistema de administración para controlar tanto los datos como los usuarios. La administración de bases de datos se realiza con un DBMS, el cual es un conjunto

de servicios (aplicaciones de software) para administrar bases de datos que permiten:

- Un fácil acceso a los datos.
- El acceso a la información por parte de múltiples usuarios.
- La manipulación de los datos encontrados en la base de datos (insertar, eliminar, editar) [15] como se presenta en la figura 6.

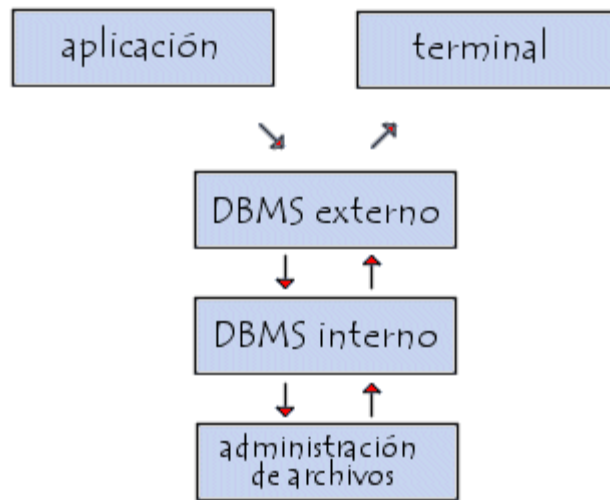


Figura 6. Representa el DBMS y como debe ser el flujo de la información.

Fuente: [15].

El DBMS puede dividirse en tres subsistemas:

- El sistema de administración de archivos:
Para almacenar información en un medio físico
- El DBMS interno:
Para ubicar la información en orden
- El DBMS externo:
Representa la interfaz del usuario [15]

2.7 Inteligencia de negocios y administración.

El conocimiento es el único factor que puede hacer que surja la compañía. La tecnología de la información como Internet ha revolucionado la forma en la economía, la publicidad, las ventas trabajo, y la información en sí puede ser valiosa. Se debe asegurar el estar en la cima de las necesidades de información de su empresa con la actividad de vigilancia y los programas de inteligencia de negocios.

2.7.1 Software de gestión de proyecto.

Este tipo de software proporciona un conjunto de herramientas que se pueden usar para planificar y gestionar un proyecto, especialmente cuando existen muchas personas que trabajan en él (Ricardo, 2009).

2.7.2 Procesamiento analítico en línea.

Es una solución utilizada en el campo de la Inteligencia de Negocios (*Business Intelligence*), la cual consiste en consultas a estructuras multidimensionales (o Cubos OLAP) que contienen datos resumidos de grandes Bases de Datos o Sistemas Transaccionales (OLTP). Se usa en informes de negocios de ventas, marketing, informes de dirección, minería de datos y áreas similares (Ricardo, 2009).

2.8 Lenguaje Unificado de Modelado

UML mejor conocido como Lenguaje Unificado de Modelado por sus siglas en ingles (*Unified Modeling Language*) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan, mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación (Booch & Rumbaugh, 2006).

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Empezó como una consolidación del trabajo de

Grade Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares (Popkin, 2005).

En 1996, el *Object Management Group* (OMG), un pilar estándar para la comunidad del diseño orientado a objetos, publicó una petición con propósito de una meta-modelo orientado a objetos de semántica y notación estándares. UML, en su versión 1.0, fue propuesto como una respuesta a esta petición en enero de 1997. Hubo otras cinco propuestas rivales. Durante el transcurso de 1997, los seis promotores de las propuestas, unieron su trabajo y presentaron al OMG un documento revisado de UML llamado UML versión 1.1. Este documento fue aprobado por el OMG en Noviembre de 1997. El OMG llama a este documento OMG UML versión 1.1. El OMG está actualmente en proceso de mejorar una edición técnica de esta especificación, prevista su finalización para el 1 de abril de 1999 (Popkin, 2005).

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real. UML en su primera versión (UML 1.0) ofrece nueve diagramas de los cuales se describen los 7 más importantes a continuación. 41

2.8.1 Diagrama de Casos de Uso

Un diagrama de casos de uso es una especie de diagrama de comportamiento. UML mejorado El Lenguaje de Modelado Unificado define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede sólo dar una vista general simple de un caso de uso o un conjunto de casos de uso. Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso (Booch & Rumb)

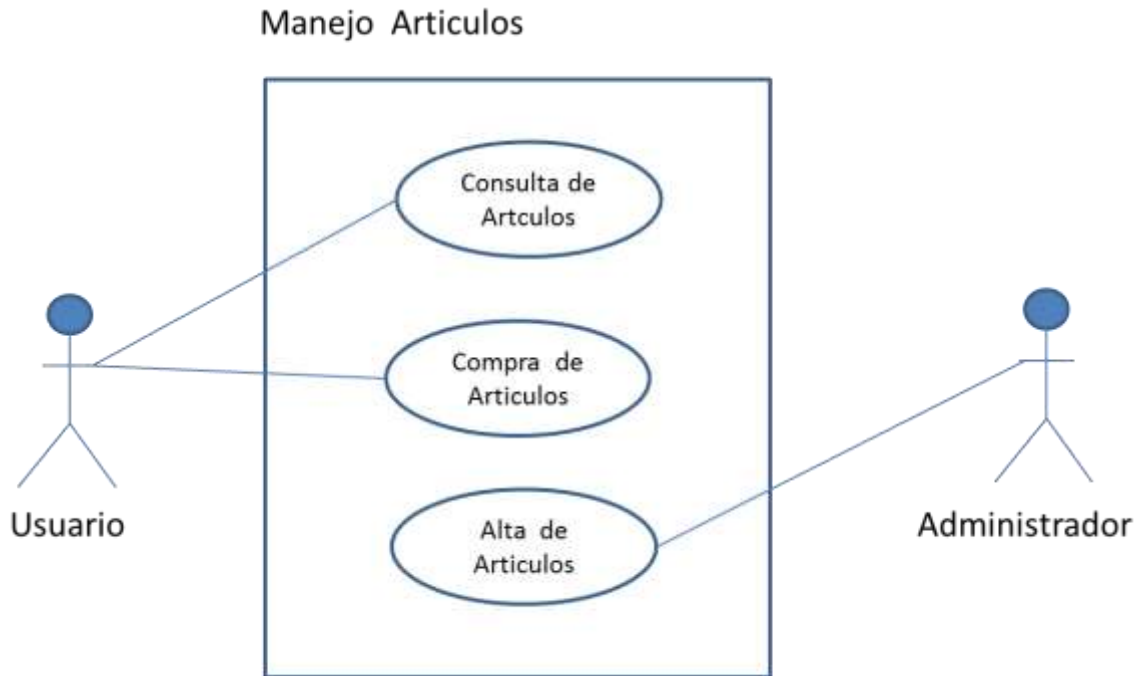


Figura 6 Ejemplo de diagrama de caso de uso.

Fuente elaboración propia

2.10 PHP& MySQL como herramienta de desarrollo

2.10.1 Breve Historia de PHP

PHP proviene de un programa anterior, llamado PHP/FI. PHP/FI fue creado por Rasmus Lerdorf en 1995, inicialmente como un simple conjunto de scripts de Perl para controlar quién accedía a su *curriculum* en línea. Llamó a ese conjunto de scripts Personal Home Page Tools. Según se requería, Rasmus fue añadiendo más funcionalidad escrita en el Lenguaje C que era capaz de comunicarse con bases de datos y permitía a los usuarios desarrollar sencillas aplicaciones Web dinámicas. Rasmus decidió volver libre el código fuente de PHP/FI para que cualquiera pudiese utilizarlo, así como arreglar errores y mejorar el código (Boronczyk, 2009).

PHP/FI, que se mantuvo para páginas personales y como intérprete de formularios, incluídas algunas de las funciones básicas de PHP tal y como lo conocemos hoy. Tenía variables como las de Perl, interpretación automática de variables de formulario y sintaxis incrustada en el HTML. La sintaxis por sí misma

era similar a la de Perl, aunque mucho más limitada, simple y algo inconsistente (Rasmus, 2011).

En 1997, PHP/FI 2.0, la segunda vez que se escribió la implementación en C, tuvo un seguimiento estimado de varios miles de usuarios en todo el mundo, aproximadamente 50,000 dominios tenían PHP/FI instalado, sumando alrededor del 1% de los dominios de Internet. Mientras había mucha gente contribuyendo con líneas de código a este proyecto, en su mayor parte era todavía el proyecto de una sola persona (Rasmus, 2011).

PHP/FI 2.0 no se liberó oficialmente hasta Noviembre de 1997, después de estar la mayoría del tiempo en versión beta. Después de poco tiempo de liberarse aparecieron las primeras versiones *alpha* de PHP 3.0 (Rasmus, 2011)

2.10.2 Lenguaje de programación interpretado PHP

PHP es un lenguaje de programación interpretado. PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools) (Boronczyk, 2009).

Un ejemplo del uso y potencia de este lenguaje son Wikipedia y la red Social Facebook, mismas que están desarrolladas en PHP. PHP Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Capítulo III Metodología.

3.1. Modelo de la Cascada.

Este modelo es a veces llamado ciclo de vida clásico, sugiere un enfoque sistemático y secuencial de pasos para el desarrollo del software el cual es el más antiguo, usado en desarrollos pequeños, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado. Ver la figura 5. (Roger S. Pressman, 2010).

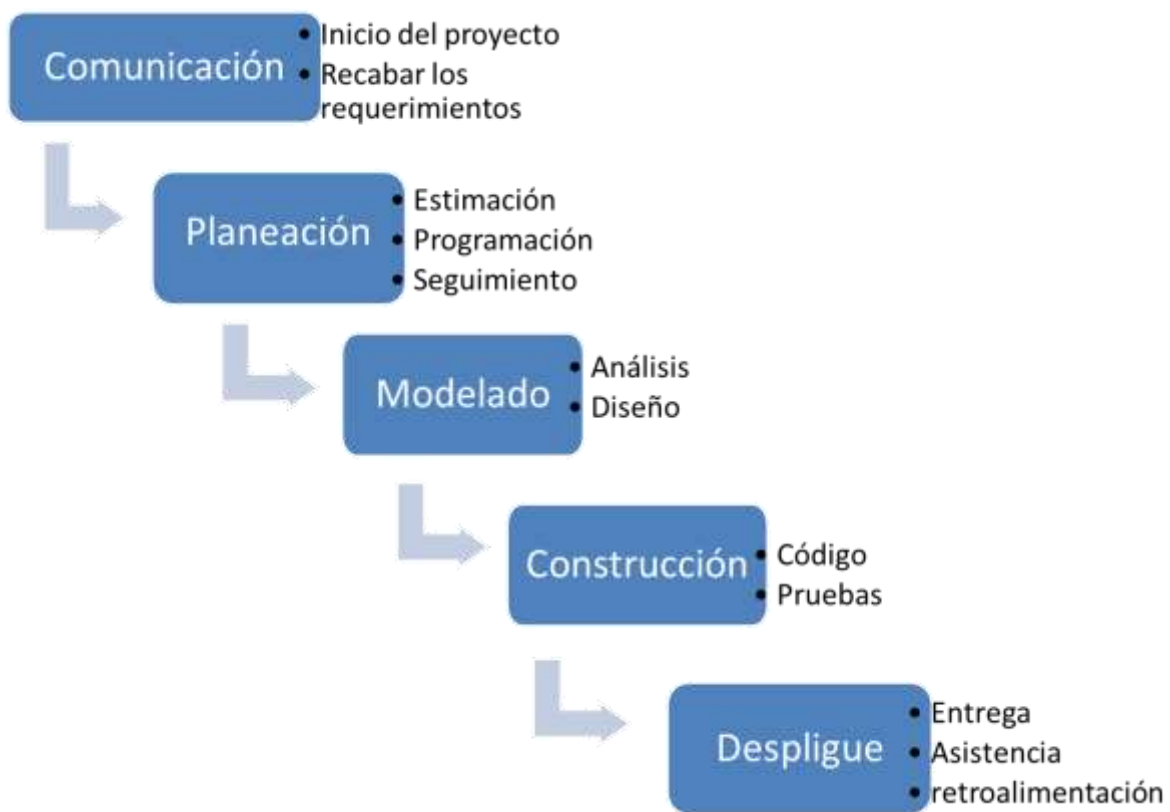


Figura 5. Modelo de la cascada
Fuente: (Roger S. Pressman, 2010).

PRINCIPIOS QUE GUÍAN TODA LA ACTIVIDAD ESTRUCTURAL

a) Principios de Comunicación.

Principios de Comunicación.	En esta fase se determina el método como se va a llevar la comunicación entre el cliente y el equipo de desarrollo, en esta fase se llevan entrevistas con el personal involucrado en el proceso que se desea automatizar, para levantar las necesidades que se desea cubrir por el software a desarrollar, estas entrevistas se realizan por preguntas, y observación a los procesos. Es importante en esta etapa detectar quien va ser el beneficiado principal.	1. Escuchar.	Trate de centrarse en las palabras del hablante en lugar de formular su respuesta a dichas palabras. Si algo no está claro, pregunte para aclararlo, pero evite las interrupciones constantes. Si una persona habla, nunca parezca usted beligerante en sus palabras o actos (por ejemplo, con giros de ojos o movimientos de la cabeza).
		2. Antes de comunicarse, prepararse.	Dedique algún tiempo a entender el problema antes de reunirse con otras personas. Si es necesario, haga algunas investigaciones para entender el vocabulario propio del negocio. Si tiene la responsabilidad de conducir la reunión, prepare una agenda antes de que ésta tenga lugar.
		3. Alguien debe facilitar la actividad.	Toda reunión de comunicación debe tener un líder (facilitador) que 1) mantenga la conversación en movimiento hacia una dirección positiva, 2) sea un mediador en cualquier conflicto que ocurra y 3) garantice que se sigan otros principios.

		4. Es mejor la comunicación cara a cara.	Pero por lo general funciona mejor cuando está presente alguna otra representación de la información relevante. Por ejemplo, un participante quizá genere un dibujo o documento en “borrador” que sirva como centro de la discusión.
		5. Tomar notas y documentar las decisiones.	Las cosas encuentran el modo de caer en las grietas. Alguien que participe en la comunicación debe servir como “secretario” y escribir todos los temas y decisiones importantes.
		6. Perseguir la colaboración.	La colaboración y el consenso ocurren cuando el conocimiento colectivo de los miembros del equipo se utiliza para describir funciones o características del producto o sistema. Cada pequeña colaboración sirve para generar confianza entre los miembros del equipo y crear un objetivo común para el grupo.
		7. Permanecer centrado; hacer módulos con la discusión.	Entre más personas participen en cualquier comunicación, más probable es que la conversación salte de un tema otro. El facilitador debe formar módulos de conversación para abandonar un tema solo después de que se haga resuelto (sin embargo, considere el principio 9).

		8. Si algo no está claro, hacer un dibujo.	La comunicación verbal tiene sus límites. Con frecuencia, un esquema o dibujo arroja claridad cuando las palabras no bastan para hacer el trabajo.
		9. Avanzar	La comunicación, como cualquier actividad de ingeniería de software, requiere tiempo. En vez de hacer iteraciones sin fin, las personas que participan deben reconocer que hay muchos temas que requieren análisis (véase el principio 2) y que “avanzar” es a veces la mejor forma de tener agilidad en la comunicación.
		10. Negociación.	Hay muchas circunstancias en las que usted y otros participantes deben negociar funciones y características, prioridades y fechas de entrega. Si el equipo ha colaborado bien, todas las partes tendrán un objetivo común. Aun así, la negociación demandará el compromiso de todas las partes.

b)Principios de Planeación.

Principios de Planeación.	Es la etapa en la que se determina el alcance del proyecto, si es o no factible de realizar además se determinan tiempos y costos aproximados, estableciendo así la ruta crítica de cada actividad. Esto es porque la falta de planeación de un sistema es la causa principal de retrasos en programación, incremento de costos, poca calidad, y altos costos de mantenimiento en los desarrollos de productos de software.	1. Entender el alcance del proyecto	Es imposible usar el mapa si no se sabe a dónde se va. El alcance da un destino al equipo de software.
		2. Involucrar en la actividad de planeación a los participantes del software	Los participantes definen las prioridades y establecen las restricciones del proyecto. Para incluir estas realidades, es frecuente que los ingenieros de software deban negociar la orden de entrega, los plazos y otros asuntos relacionados con el proyecto.
		3. Reconocer que la planeación es iterativa.	Un plan para el proyecto nunca está grabado en piedra. Para cuando el trabajo empiece, es muy probable que las cosas hayan cambiado. En consecuencia, el plan deberá ajustarse para incluir dichos cambios. Además, los modelos de proceso interactivo incrementales dictan que debe repetirse la planeación después de la entrega de cada incremento de software, con base en la retroalimentación recibida de los usuarios.
		4. Estimar con base en lo se sabe.	El objetivo de la estimación es obtener un índice del esfuerzo, costo y duración de las tareas, con base en la comprensión que tenga el equipo sobre el trabajo que va a realizar. Si

			la información es vaga o poco confiable, entonces las estimaciones tampoco serán confiables.
		5. Al definir el plan, tomar en cuenta los riesgos.	Si ha identificado riesgos que tendrían que tendrán un efecto grande y es muy probable que ocurran, entonces es necesario elaborar planes de contingencia.
		6. Ser realista	Las personas no trabajan 100% todos los días. En cualquier comunicación humana hay ruido. Las omisiones y ambigüedades son fenómenos de la vida. Los cambios ocurren. Aun los mejores ingenieros de software cometen errores. Estas y otras realidades deben considerarse al establecer un proyecto.
		7. Ajustar la granularidad cuando se defina el plan.	Se refiere al nivel de detalle que se adopta cuando se desarrolla un plan. En general, la granularidad va de poca a mucha conforme el tiempo avanza.
		8. Definir cómo se trata de asegurar la calidad	El plan debe identificar la forma en la que el equipo de software busca asegurar la calidad. Si se realizan revisiones técnicas, deben programarse.
		9. Describir cómo se busca	Aun la mejor planeación puede ser anulada por el cambio sin control. Debe identificarse la

		manejar el cambio	forma en la que van a recibirse los cambios a medida que avanza el trabajo de la ingeniería de software.
		10. Dar seguimiento al plan con frecuencia y hacer los ajustes que se requieran.	Evaluar diariamente el avance, en busca de áreas y situaciones problemáticas en las que las actividades programadas no se apeguen al avance real. Cuando se detecten desviaciones, hay que ajustar el plan en consecuencia.

c) Principios de Modelado.

Principios de Modelado.	Se crean modelos para entender mejor la entidad real que se va a construir. En el trabajo de ingeniería de software se crean dos clases de modelos: de requerimiento y de diseño. Los modelos de requerimiento (también conocidos como modelos de análisis) representan los requerimientos del cliente mediante la ilustración del software en tres dominios diferentes: el de la información, el funcional y el de comportamiento. Los modelos de diseño representan características del software que ayudan a los profesionales a elaborarlo con eficacia: arquitectura, interfaz de usuario y detalle en el nivel de componente	1. El equipo de software tiene como objetivo principal elaborar software no crear modelos.	Agilidad significa entregar software al cliente de la manera más rápida posible. Los modelos que contribuyan a esto son beneficios, pero deben evitar aquellos que hagan lento el proceso o que den poca perspectiva.
		2. Viajar ligero, no crear más modelos de los necesarios.	Todo modelo que se cree debe actualizarse si ocurren cambios. Mas importante aún es que todo modelo nuevo exige tiempo, que de otra manera se destinaria a la construcción (codificación y pruebas). Entonces, cree sólo aquellos modelos que hagan más fácil y rápido construir el software.
		3. Tratar de producir el modelo más sencillo que describa al problema o al software.	No construya software en demasía. Al mantener sencillos los modelos, el software también lo será. El resultado es que tendrá un software fácil de integrar, de probar y de mantener. Además los modelos sencillos son fáciles de entender y criticar, lo que da como resultado un formato funcional de retroalimentación que optimiza el resultado final.
		4. Construir	Suponga que sus modelos cambiarán, pero vigile

		modelos susceptibles al cambio	que esta suposición no lo haga descuidado.
		5. Ser capaz de enunciar un propósito explícito para cada modelo que se cree.	Cada vez que se cree un modelo, pregúntese por qué lo hace. Si no encuentra una razón sólida para la existencia del modelo. No pierda tiempo en él.
		6. Adaptar los modelos que se desarrollan al sistema en cuestión.	Tal vez sea necesario adaptar a la aplicación la notación del modelo o las reglas; por ejemplo, una aplicación de juego de video quizá requiera una técnica de modelado distinta que el software incrustado que controla el motor de un automóvil en tiempo real.
		7. Tratar de construir modelos útiles, pero olvidarse de elaborar modelos perfectos	Cuando se construyen modelos de requerimientos y diseño, alcanza un punto de rendimientos decrecientes. Pero el modelo debe hacerse con la mirada puesta en las siguientes etapas de la ingeniería de software. Las iteraciones sin fin para obtener un modelo “perfecto” no cumplen la necesidad de agilidad.

		8. No ser dogmático respecto de la sintaxis del modelo. Si se tiene éxito para comunicar contenido, la representación es secundaria.	Aunque cada miembro del equipo de software debe tratar de usar una notación consistente durante el modelado, la característica más importante del modelo es comunicar información que permita la realización de la siguiente tarea de ingeniería. Si un modelo tiene éxito en hacer esto, es perdonable la sintaxis incorrecta.
		9. Si su instinto dice que un modelo no es correcto a pesar de que se vea bien en el papel, hay razones para estar preocupado.	Confíe en su instinto. Si algo dice que el modelo de diseño está destinado a fracasar (aun cuando esto no pueda demostrarse en forma explícita), hay razones para dedicar más tiempo a su estudio o a desarrollar otro distinto.
		10. Obtener retroalimentación tan pronto	Todo modelo debe ser revisado por los miembros del equipo. El objetivo de estas revisiones es obtener retroalimentación para utilizarla a fin de

		como sea posible.	corregir los errores de modelado, cambiar las interpretaciones equivocadas y agregar las características o funciones omitidas inadvertidamente.
--	--	-------------------	---

d)Principios de Construcción.

Principios de construcción	Los principios que guían el trabajo de codificación se relacionan de cerca con el estilo, lenguajes y métodos de programación. Sin embargo puede enunciarse cierto número de principios fundamentales	Principios de preparación: Antes de escribir una sola línea de código, asegúrese de:	• Entender el problema que se trata de resolver. • Comprender los principios y conceptos básicos del diseño. • Elegir un lenguaje de programación que satisfaga las necesidades del software que se va a elaborar y el ambiente en el que operará. • Seleccionar un ambiente de programación que disponga de herramienta que hagan más fácil su trabajo. • Crear un conjunto de pruebas unitarias que se aplicarán una vez que se haya terminado el componente a codificar.
		Principios de	• Restringir sus algoritmos por medio

		programación: Cuando comience a escribir código, asegúrese de:	del uso de programación estructurada. • Tomar en consideración el uso de programación por parejas. • Seleccionar estructuras de datos que satisfagan las necesidades del diseño. • Entender la arquitectura del software y crear interfaces que son congruentes con ella. • Mantener la lógica condicional tan sencilla como sea posible. • Crear lazos anidados en forma tal que se puedan probar con facilidad. • Seleccionar nombres significativos para variables y seguir otros estándares locales de codificación. • Escribir código que se documente a sí mismo. • Crear una imagen visual (por ejemplo:
--	--	--	---

			líneas con sangría y en blanco) que ayuden a entender.
		Principio de validación: Una vez que haya terminado su primer intento de codificación, asegúrese de:	• Realizar el recorrido del código cuando sea apropiado. • Llevar a cabo pruebas unitarias y corregir los errores que se detecten. • Rediseñar el código.
		Principios de la Prueba La prueba es el proceso que ejecuta un programa con objeto de encontrar un error.	
		1. Todas las pruebas deben poder rastrearse hasta los requerimientos del cliente.	El objetivo de las pruebas de software es descubrir errores. Entonces, los defectos más severos (desde el punto de vista del cliente) son aquellos que hacen que el programa no cumpla sus requerimientos.
		2. Las pruebas deben planearse	La planeación de las pruebas comienza tan pronto como se termina el modelo de requerimientos. La definición detallada de

		mucho antes de que den comienzo.	casos de prueba principia apenas se ha concluido el modelo de diseño. Por tanto, todas las pruebas pueden planearse y diseñarse antes de generar cualquier código.
		3. El principio de Pareto se aplica a las pruebas de software.	En este contexto, el principio de Pareto implica que 80% de todos los errores no detectados durante las pruebas se relacionan con 20% de todos los componentes de programas. Por supuesto, el problema es aislar los componentes sospechosos y probarlos a fondo.
		4. Las pruebas deben comenzar “en lo pequeño” y avanzar hacia “lo grande”.	Las primeras pruebas planeadas y ejecutadas por lo general se centran en componentes individuales. Conforme avanzan las pruebas, la atención cambia en un intento por encontrar errores en grupos integrados de componentes y, en última instancia, en todo el sistema.
		5. No son posibles	Hasta para un programa de tamaño

		las pruebas exhaustivas	moderado, el número de permutaciones de las rutas es demasiado grande. Por esta razón, durante una prueba es imposible ejecutar todas las combinaciones de rutas. Sin embargo, es posible cubrir en forma adecuada la lógica del programa y asegurar que se han probado todas las condiciones en el nivel de componentes.
--	--	--------------------------------	--

e) Principios de Despliegue.

Principios de construcción	La entrega de un incremento de software representa un punto de referencia importante para cualquier proyecto de software. Cuando el equipo se prepara para entregar un incremento, deben seguirse ciertos principios clave.	1. Deben manejarse las expectativas de los clientes.	No se debe prometer más de lo que puede entregar en un plazo previsto, o entregar más de lo que se prometió en un incremento de software y para el siguiente menos, esto puede ocasionar la desilusión por parte del cliente y por consecuencia la retroalimentación no sea productiva.
		2. Debe ensamblarse y probarse el paquete completo que se entregara.	Todos los scripts de instalación y otras características de operación deben ejecutarse por completo en tantas configuraciones diferentes de computo como sea posible.
		3. Antes de entregar el software, debe establecer un régimen de	Si el apoyo es adhoc, o peor aún, no existe, el cliente quedará insatisfecho de inmediato. El apoyo debe planearse, el materia respectivos deben prepararse y los mecanismos

		apoyo.	apropiados de registro deben establecerse a fin de que el equipo de software realice una evaluación una evaluación categórica de las clases de apoyo solicitado.
		4. Se deben proporcionar a los usuarios finales materiales de aprendizaje apropiados	El equipo de software debe desarrollar materiales de capacitación apropiados; es necesario proveer lineamientos para solución de problemas y cuando sea necesario, debe publicarse “lo que es diferente en este incremento de software”.
		5. El software defectuoso debe corregirse y después entregarse.	Los clientes olvidarán pronto que entregaste un producto de alta calidad con la advertencia de que los errores “se corregirán en la siguiente entrega”. Esto es un error.

Es importante recordar que el software entregado brinda beneficios al usuario final, pero también da retroalimentación útil para el equipo que lo desarrolló. Cuando el incremento se libere, debe invitarse a los usuarios finales a que comenten acerca de características y funciones, facilidad de uso, confiabilidad y cualesquiera otras características (Roger S. Pressman, 2010).

3.1.1. Comunicación.

Antes de que inicie cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto al proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software (Roger S. Pressman, 2010).

3.1.2. Planeación.

En un proyecto la actividad de planeación crea un “mapa” que guía al equipo mientras viaja. El mapa denominado plan del proyecto de software define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades (Roger S. Pressman, 2010).

Para lograr el éxito del proyecto se definen las siguientes actividades que se ejecutarán durante esta etapa:

3.1.2.1. Estudio de factibilidad:

Su finalidad es la de comprobar la factibilidad de desarrollar el sistema en los aspectos: técnico, operativo y económico mismos que se describen a continuación (Roger S. Pressman, 2010):

a) Factibilidad técnica:

Esta parte del estudio consistirá en proponer las especificaciones técnicas necesarias para el sistema, y así evaluar si el laboratorio de cómputo de la UABC cuenta y cumple con los requerimientos tecnológicos necesarios para el desarrollo del sistema, se dividirá en: Hardware y software.

b) Factibilidad operativa:

Esta parte del estudio pretende evaluar si existe resistencia al cambio por parte del personal que labora en el laboratorio de cómputo de la UABC, si aceptarán o no la implantación de un sistema de información para el apoyo de las actividades que se realizan.

c) Factibilidad económica:

En esta parte del estudio se evaluará el aspecto económico, si se cuentan con los recursos económicos para llevar a cabo un proyecto de desarrollo de software, los recursos disponibles no monetarios como equipo e infraestructura con la que ya cuentan en la facultad, para así determinar el costo neto del desarrollo del sistema. Cabe mencionar que este es un proyecto con fines académicos, por lo que se busca aprovechar todo los recursos ya disponibles, además de la utilización de tecnologías libres que no representen un costo en el desarrollo.

d) Cronograma de actividades:

Una vez que se compruebe que es factible el desarrollo del sistema, se definen y programan las actividades necesarias que se llevarán a cabo durante las siguientes etapas del ciclo de desarrollo.

Por último es importante tener en cuenta que con frecuencia se dice que es imposible realizar una planeación inicial, porque la información precisa sobre las metas del proyecto, necesidades del cliente y restricciones del producto no se conocen al comenzar el proyecto de desarrollo, sin embargo, uno de los principales propósitos de esta fase es aclarar los objetivos, problemas o necesidades y restricciones. La dificultad de la planeación no debe desalentar tan importante actividad.

3.1.3. Modelado.

Se crean los modelos para entender de mejor forma los requerimientos del software y el diseño que los satisfaga (Roger S. Pressman, 2010).

3.1.3.1. Análisis.

En esta etapa también conocida como pre-análisis, se determinan las condiciones actuales del laboratorio de cómputo de la UABC, ubicando los puntos críticos, así como los elementos fundamentales de los procedimientos usados, localizando los principales problemas a resolver. Es indispensable comprender perfectamente los requisitos del software, para que éste no fracase. Esta etapa puede parecer una tarea relativamente sencilla, pero las apariencias engañan ya que abundan los casos en que se puede llegar a falsas interpretaciones o con falta de información.

Para lograr esto se realizarán entrevistas con el encargado del laboratorio de cómputo de la UABC, con el objetivo de determinar la situación actual y por lo tanto identificar las necesidades, mismas que serán escritas en el levantamiento de requerimientos funcionales y no funcionales los cuales servirán de punto de partida para la etapa de diseño.

3.1.3.2. Diseño

El diseño del software es realmente un proceso que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la construcción. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

Para lograr ejecutar esta etapa se hará uso del **estándar UML** para diseñar el modelo del sistema basándose en los requerimientos obtenidos durante la etapa de análisis: a través de los diagramas de **casos de uso** en los que modelarán gráficamente estos requerimientos identificando los actores y como éstos interactúan con el sistema.

También se realizarán los bosquejos de la **interfaz de usuario**, la cual describirá los aspectos estéticos, la distribución de los componentes propios de una interfaz de usuario y donde también se mostrará un acercamiento a la

funcionalidad del sistema, dicha interfaz será presentada a usuarios finales para obtener retroalimentación y así realizar los ajustes necesarios.

Otro de los aspectos que se trata en la etapa de diseño es la de modelar las estructuras de datos que manejará el sistema, para lograr esto primero se inicia utilizando los diagramas de flujo de datos para conocer el flujo de la información dentro del sistema. Después de esto se está también en condición de definir el menú de opciones el cual definirá la navegabilidad que tendrán los usuarios dentro del sistema. Y por último, esta etapa culmina con el modelado de la estructura de base de datos utilizando los diagramas **entidad-relación** mismos que servirán durante la etapa siguiente del ciclo de desarrollo.

3.1.4. Construcción.

Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste (Roger S. Pressman, 2010).

3.1.4.1. Prueba

Una vez que se ha generado el código, sigue la prueba del sistema. La verificación se centra en la lógica interna del software, asegurando que todas las sentencias se han probado y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren. Para lograr esto, el sistema se somete a pruebas de uso por parte del desarrollador y usuarios del sistemas (maestros), con el propósito de identificar errores y cambios, para así realizar los ajustes últimos y contar con un sistema final que cumpla con todos los requerimientos y lo más importante sea un producto de software de calidad ya que será el producto de la aplicación de toda una metodología ampliamente probada en el mundo de desarrollo de la ingeniería del software.

3.1.5. Despliegue.

El software (Como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación (Roger S. Pressman, 2010).

3.1.5.1. Mantenimiento

Después de ejecutar la etapa de pruebas es indudable que el software una vez entregado al cliente pueda requerir cambios en el algún momento determinado, esto se puede deber a diversos factores entre los que destacan la evolución natural de las organizaciones, por lo que el sistema tendría que adaptarse a las nuevas necesidades. Otro de los factores es que pueda llegar a identificarse algún error que no haya sido descubierto durante la etapa de pruebas, o también que éste sea causado por una situación ajena al desarrollo del sistema como pueden ser: Fallas en el servidor o fallas en la infraestructura de la red de la UABC, entre otras.

Es importante mencionar que esta etapa puede ejecutarse indefinidamente y esto se demuestra en lo que se mencionó anteriormente, de que un sistema puede requerir cambios operativos en función de la evolución de las instituciones o de otros factores pragmáticos, por lo tanto al ser un proyecto académico donde su duración está determinada por la duración en la que el desarrollador determine. Sin embargo se dejarán definidas las recomendaciones pertinentes para el mantenimiento del sistema y así disminuir en lo posible que puedan presentarse eventualidades que podrían tener un impacto negativo en la funcionalidad del sistema.

3.2. Sistema

Una vez ejecutadas todas las etapas de la metodología se contará con producto de desarrollo de software en un sistema de información funcional y de calidad, fruto de la aplicación de una metodología perteneciente a la ingeniería del software (Roger S. Pressman, 2010).

Capítulo IV. Desarrollo

4.1. Arquitectura de Software

Al hablar del desarrollo de software se debe mencionar la arquitectura en la cual se basará. La arquitectura del Software es la organización fundamental del sistema que incluye a sus componentes, sus relaciones entre ellos y el ambiente, así como de los principios que dictan su diseño y evolución [3].

La arquitectura de software también involucra:

- Funcionalidad
- Usabilidad
- Tolerancia a cambios
- Desempeño
- Reutilización
- Restricciones económicas y tecnológica
- Aspectos estéticos

La arquitectura del software desarrollado es **cliente/servidor** y hace uso de la red de Internet para poder acceder a él, ya sea como usuario (maestro) o como administrador.

En la figura 7 se muestra el patrón de la arquitectura “Cliente delgado/Servidor grueso”, que consiste en que el **cliente** hace uso de un navegador para el envío de peticiones y en el **servidor** se procesan para su pronta respuesta [4].

Las ventajas de éste tipo de arquitectura es la centralización del control de acceso a la información y fácil mantenimiento. Una de las grandes desventajas es la potencial congestión del tráfico generado por los clientes, por ésta razón se debe contar con un servidor multi-usuario que tenga bastante capacidad de procesamiento y una conexión rápida a Internet.

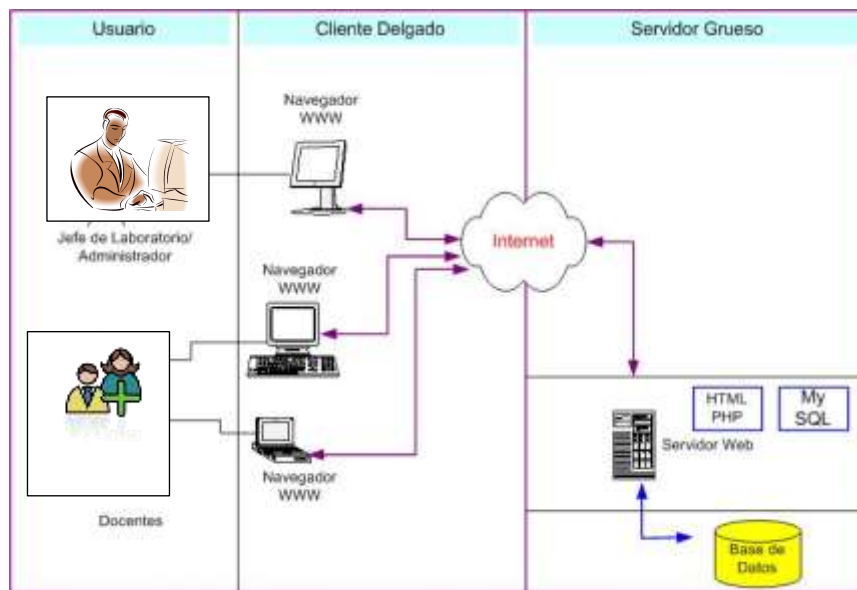


Figura 7. Patrón Cliente delgado/Servidor grueso.

Fuente elaboración propia

4.2. Diseño de base de datos

Una base de datos relacional es un conjunto finito de relaciones donde cada una de ellas contiene un esquema relacional y un cuerpo correspondiente. Los nombres de las relaciones deben ser distintivos y, dentro de cada relación, los nombres de los atributos también deben ser distintivos [5].

Los diagramas **entidad-relación** (ER) constituyen una notación para documentar un diseño tentativo de base de datos. Los analistas los utilizan para facilitar el proceso de diseño. Un diagrama entidad-relación, junto con otra documentación, capta las características importantes de la aplicación, pero su traducción legible para la máquina, que es el esquema de base de datos, en realidad lleva la información a los sistemas manejadores de base de datos (DBMS). En la figura 8 se muestra el diagrama entidad-relación creado para representar la base de datos utilizada para el sistema de gestión del centro de cómputo.

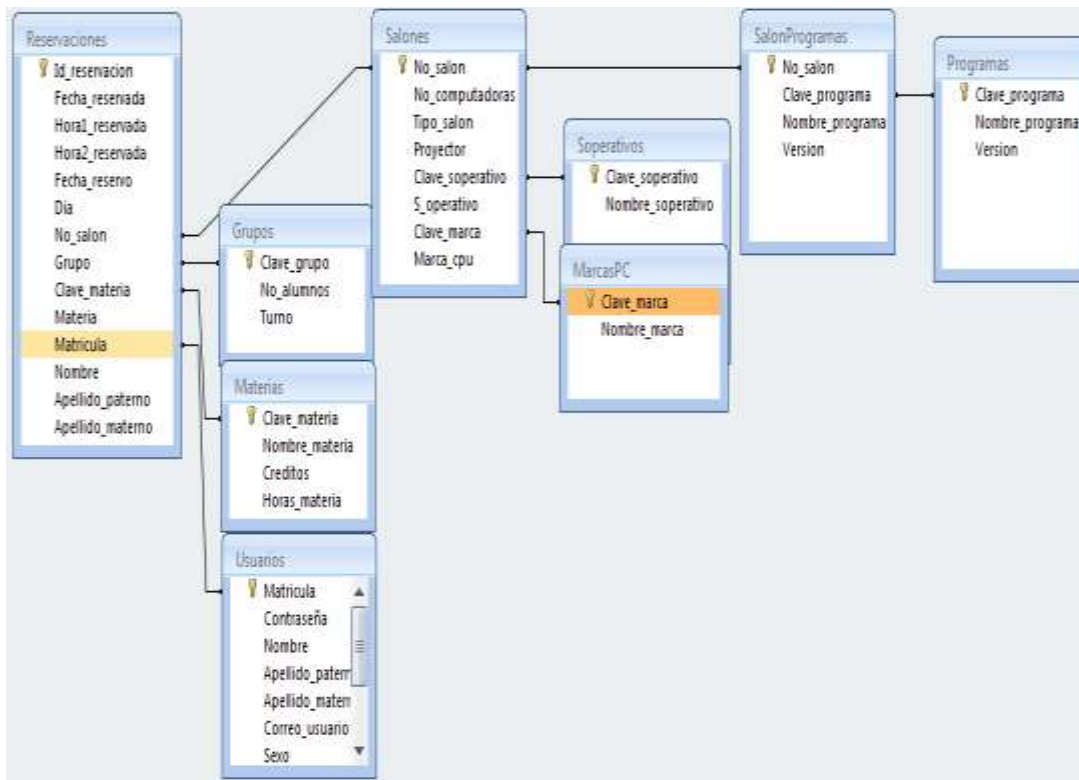


Figura 8. Diagrama entidad relación.

Fuente elaboración propia

4.3. Casos de Uso.

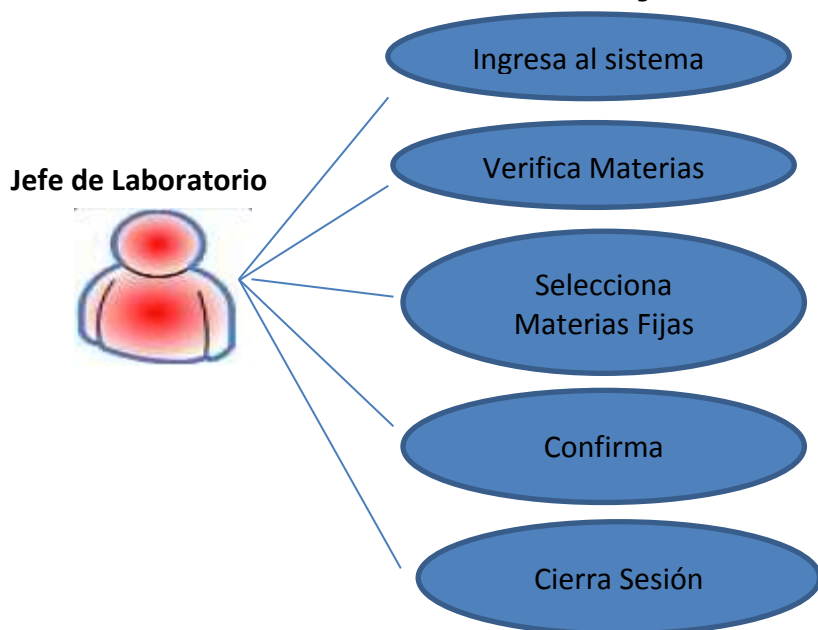
Un **caso de uso** es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de

casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo [16].

En este desarrollo es importante el emplear estos diagramas para describir procesos para que sea más sencillo entenderlos y validarlos. A continuación se presentan los casos de uso que se consideran:

- Caso de Uso #1 Selección de Materias Fijas
- Caso de Uso #2 Solicitud de Horas
- Caso de Uso #3 Reservación
- Caso de Uso #4 Asignación de Horarios

Caso de Uso #1 Selección de Materias Fijas



Detalle de los Casos de Uso

Caso de uso #1	
Nombre	Selección de Materias Fijas
Actor	Jefe de Laboratorio
Actividad	Revisar materias con clase en Laboratorio
Descripción	Este caso de uso comienza cuando el Jefe de Laboratorio ingresa al sistema de Docentes a revisar las materias que tienen por default horas en el Laboratorio para darles preferencia y asignarle un salón.

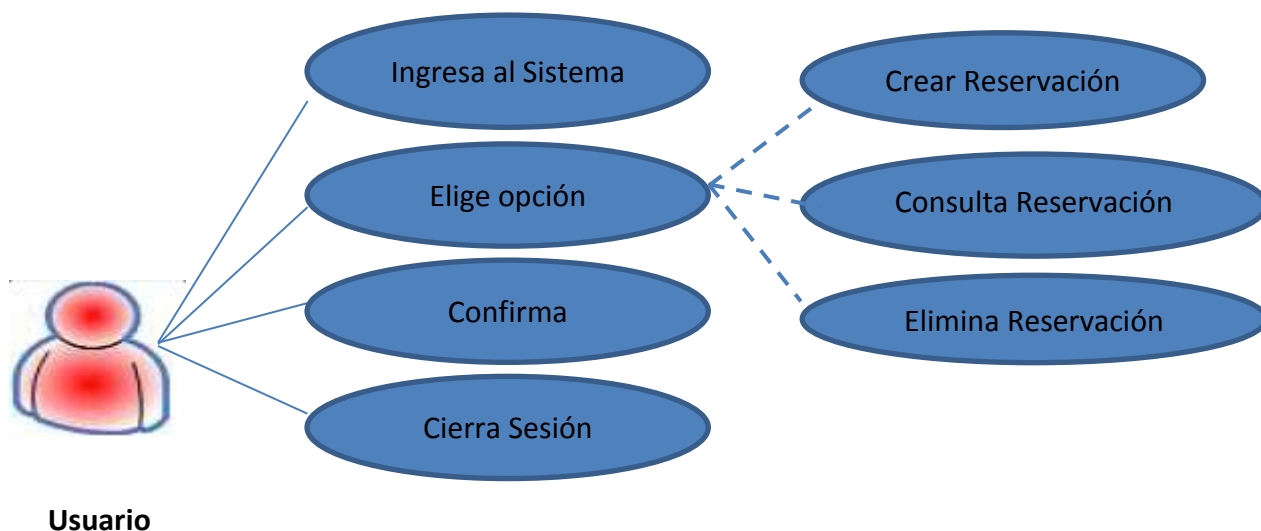
Caso de Uso #2 Solicitud de Horas



Detalle de los Casos de Uso

Caso de uso #2	
Nombre	Solicitud de Horas
Actor	Usuario
Actividad	Contar con un Laboratorio
Descripción	Este caso de uso comienza cuando el Usuario después de revisar las materias y verificar sus grupos hace una petición al Jefe de Laboratorio para que le asignen un Laboratorio por default.

Caso de Uso #3 Reservación



Detalle de los Casos de Uso

Caso de uso #3	
Nombre	Reservación
Actor	Usuario
Actividad	Hacer una Reservación de Laboratorio en el sistema
Descripción	Este caso de uso comienza cuando el Usuario ya realizó la petición de un Laboratorio y entregó el papeleo. Este ingresa al sistema y hace una reservación de un Laboratorio disponible.

Caso de Uso #4 Asignación de Horarios



Detalle de los Casos de Uso

Caso de uso #4	
Nombre	Asignación de Horarios
Actor	Jefe de Laboratorio
Actividad	Verificación de Laboratorios
Descripción	Este caso de uso comienza cuando el Jefe de Laboratorio ingresa al sistema de Reservaciones y realiza altas, bajas y consultas en el sistema (dado caso que lo requiera) sobre los laboratorios que previamente hayan seleccionado los Usuarios, confirmando así el Horario Oficial para empezar a dar Clases.

4.4. Software de gestión

Para poder lograr una eficiente administración se ha creado una diversidad de programas que ayudan con esta tarea, sus aplicaciones van desde el control de acceso a los equipos, la asignación de espacios, resguardo de equipos y software, apoyo a usuarios y las diferentes actividades que realiza un centro de cómputo. Estos programas existen tanto en forma abierta como con un costo inherente o también se han creado a la medida por personal que labora en estos departamentos [6]; tal es el caso de la F.C.A. donde la administración del laboratorio desarrolló el software que describiremos en las siguientes secciones.

4.4.1. Descripción de Software de gestión de laboratorios.

4.4.1.1. Inicio de Sección del usuario:

En la pantalla que se muestra en de la figura 9 solicita al usuario que ingrese su número de empleado en el campo de “usuario” y la contraseña en el campo correspondiente para que el sistema los valida y permita acceso al sistema. Le sigue a esto una sección de avisos que el administrador del sistema puede usar para informar a los usuarios de algún evento en el laboratorio o posible suspensión del mismo por diversas actividades por dar un ejemplo de los avisos.



Figura 9. Pantalla de inicio, Fuente elaboración propia.

A continuación, en la figura 10 se muestra la Pantalla de avisos y la visualización del menú de opciones, que están listadas del lado izquierdo de la pantalla del usuario.



Figura 10. Pantalla de Anuncios, Fuente elaboración propia

4.4.1.2. Módulo de reservación de salones para el docente:

En éste módulo el docente puede seleccionar realizar reservaciones, consultas o eliminaciones por salón ingresando la fecha y horario.

En la pantalla de la figura 11, al lado izquierdo se ve la parte de navegación donde se puede acceder a reservaciones, grupos, salones e información del usuario que accedió con su usuario y contraseña. En la figura 11 se tiene seleccionada la opción de reservaciones, en el panel del lado derecho se muestran las acciones que puede realizar en cada opción en este caso es; Reservar los Salones, Consultar y Eliminar las reservaciones, la tercera es consultar la disponibilidad de salones en cierta fecha.

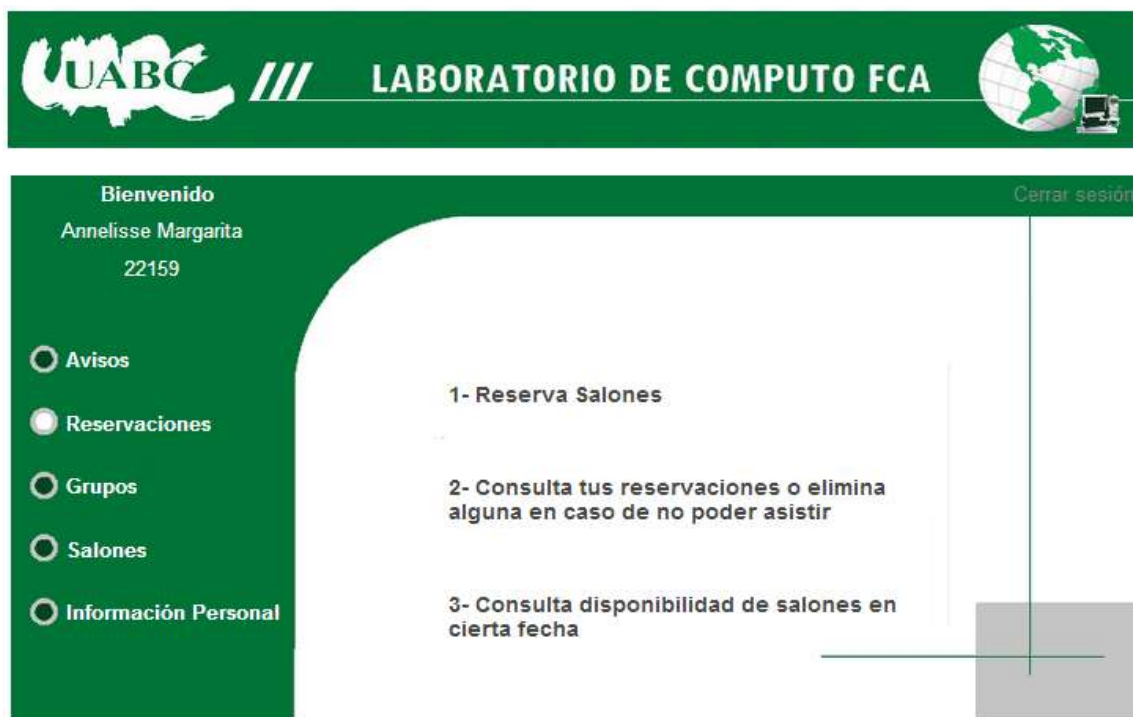


Figura 11. Módulo de reservación de salones para el docente.

Para Poder realizar una reservación el primer paso es seleccionar la fecha cuando se desea realizar la reservación del salón como se muestra en las figuras 12 y 13.



22159 Annelisse Margarita

[Inicio](#)

[Cerrar sesión](#)

Hola **Annelisse Margarita**, por favor selecciona la fecha en la cual deseas hacer una reservación.

Primero da clic sobre la imagen, al darle clic aparecerá un calendario donde seleccionaras la fecha requerida. Una vez seleccionada la fecha oprime el botón enviar.

Fecha: 

Figura 12. Se solicita la fecha de reservación.

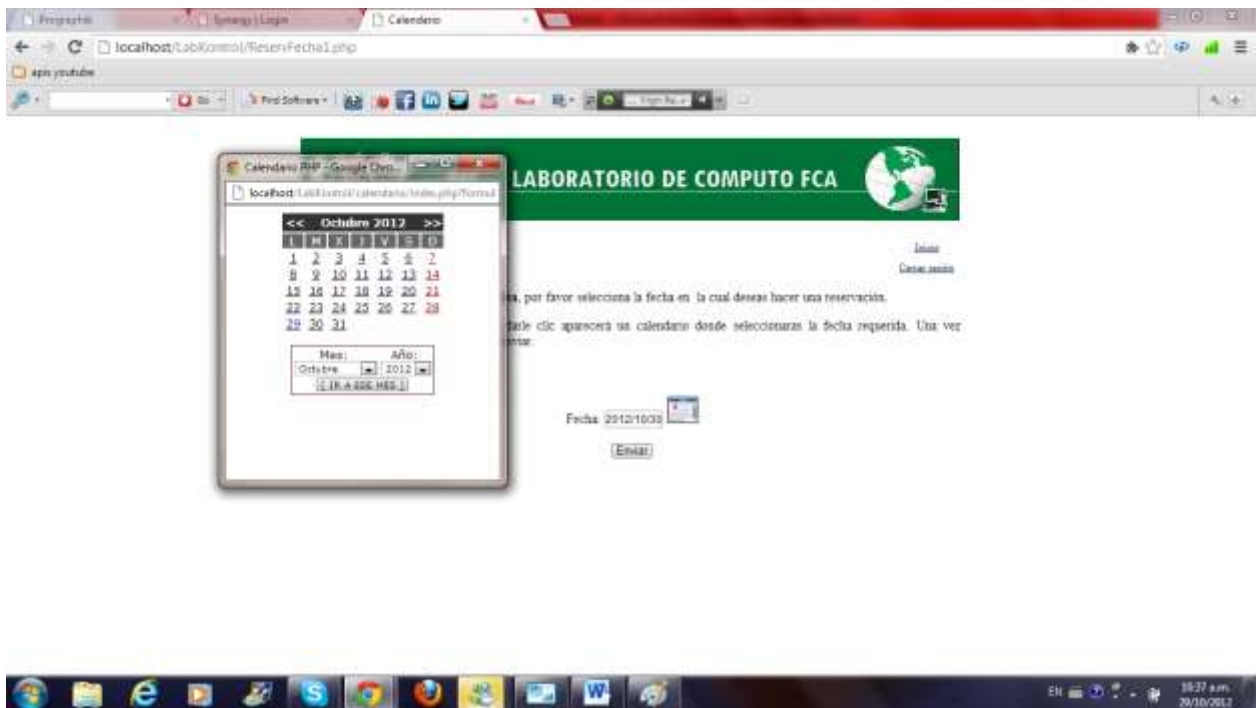


Figura 13. Se selecciona la fecha del calendario.

Se confirman los datos que el usuario ingresó para continuar el proceso, si no cambiar para seleccionar los correctos.



LABORATORIO DE COMPUTO FCA

22159 Annelisse Margarita

[Inicio](#)
[Cerrar sesión](#)

Annelisse Margarita , ha seleccionado la fecha " 2012/10/30 " , la cual equivale al día " **Martes** " .

Si la fecha que selecciono ha sido confirmada por favor seleccione **Continuar**.

Si desea cambiar la fecha seleccione **Cambiar**.

Figura 14. Confirmación de datos: fecha, día y nombre de maestro.

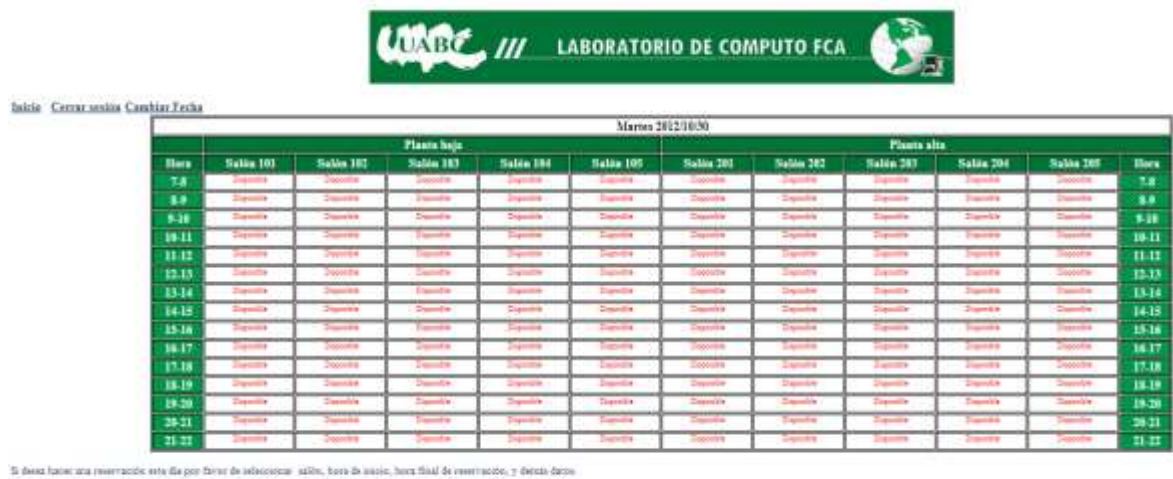


Figura 15. Se muestra la disponibilidad por salón y hora.

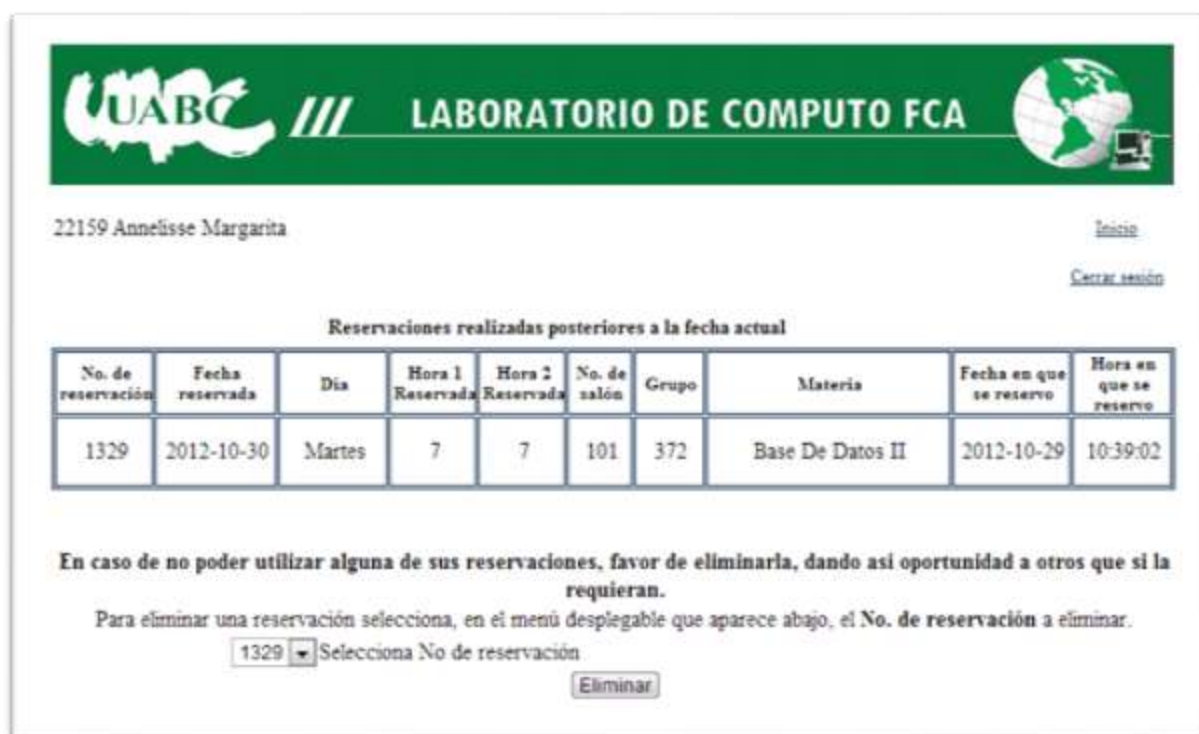


Figura 16. Se selecciona el grupo y los datos de materia.

4.4.1.2.1. Eliminación de reservaciones

Un punto importante es el proceso de los módulos es poder borrar las reservaciones por el maestro que reservó y no va poder utilizar para que ese espacio lo pueda utilizar otro maestro y sea óptimo el uso de los recursos del laboratorio de computo.

En la figura 17, se muestra un ejemplo de las reservaciones después de dar click en la opción de consulta y eliminación de reservaciones, los datos de las reservaciones activas que no han pasado en fecha en la cual se consulta para si esta correcto si no es asi la elimine y pueda realizar otra, así también como poder eliminar la que no va utilizar y quede disponible para otro maestro. Para eliminar una reservación en la pantalla que se muestra en la figura 17, seleccionas el número de reservación y le das clic en el botón de eliminar posterior a esto te parece la pantalla de la figura 18 donde te muestra los datos de la reservación eliminada.



22159 Annelisse Margarita

[Inicio](#)

[Cerrar sesión](#)

Reservaciones realizadas posteriores a la fecha actual

No. de reservación	Fecha reservada	Día	Hora 1 Reservada	Hora 2 Reservada	No. de salón	Grupo	Materia	Fecha en que se reservo	Hora en que se reservo
1329	2012-10-30	Martes	7	7	101	372	Base De Datos II	2012-10-29	10:39:02

En caso de no poder utilizar alguna de sus reservaciones, favor de eliminarla, dando así oportunidad a otros que si la requieran.

Para eliminar una reservación selecciona, en el menú desplegable que aparece abajo, el No. de reservación a eliminar.

1329 Selecciona No de reservación

Figura 17. Pantalla de muestras las reservaciones y eliminación.

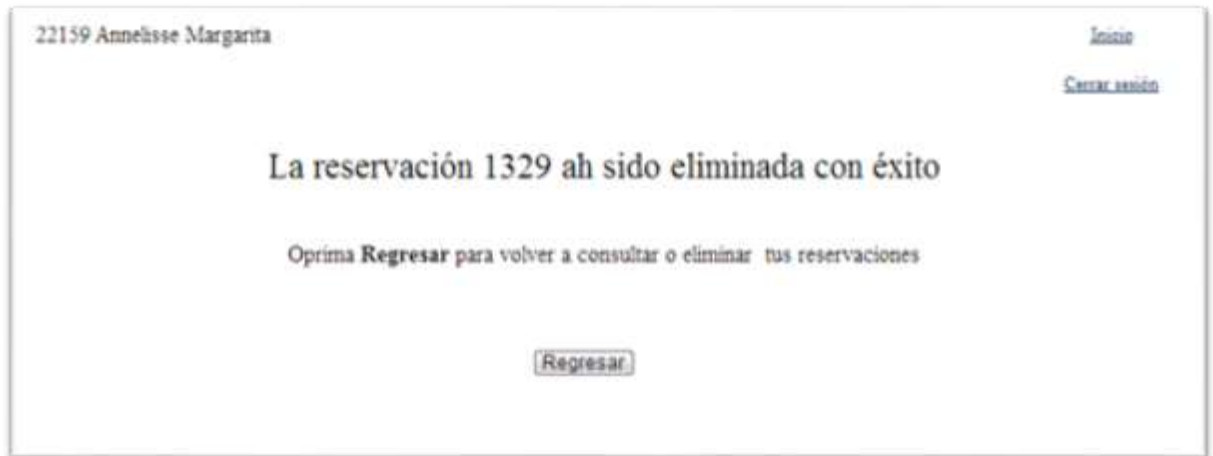


Figura 18. Muestra los datos de la reservación eliminada.

4.4.1.2.2. Consultar reservaciones

En este módulo el usuario puede realizar **consultas de disponibilidad** de espacios en salones, para su posterior reservación si lo cree conveniente. La secuencia del proceso se muestra en las pantallas de las figuras 19, 20, 21.



Figura 19. Selecciona la fecha que se desea consultar la disponibilidad.

22159 Annelisse Margarita Inicio
Cerrar sesión

Annelisse Margarita , ha seleccionado la fecha " 2011/3/16 " , la cual equivale al día " Miércoles ".

Si la fecha que selecciono ha sido confirmada por favor seleccione **Continuar**.

Si desea cambiar la fecha seleccione **Cambiar**.

Figura 20. Muestra los datos seleccionados para verificar y continuar.

///
LABORATORIO DE COMPUTO FCA

Inicio Cerrar sesión Cambiar fecha Ir a profesor

Miércoles											
Planta baja						Planta alta					
Hora	Salón 101	Salón 102	Salón 103	Salón 104	Salón 105	Salón 201	Salón 202	Salón 203	Salón 204	Salón 205	Hora
7-8	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	7-8
8-9	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	8-9
9-10	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	9-10
10-11	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	10-11
11-12	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	11-12
12-13	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	12-13
13-14	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	13-14
14-15	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	14-15
15-16	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	15-16
16-17	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	16-17
17-18	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	17-18
18-19	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	18-19
19-20	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	19-20
20-21	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	20-21
21-22	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	vacante	21-22

Figura 21. El resultado de la consulta.

4.4.1.3. Módulo de grupos:

Éste módulo sirve para que el usuario consulte el número de alumnos y turno del grupo, la secuencia de selección de grupo se muestra en las figuras 22 y 23, observe un desplegado de información en la figura 24.



Figura 22. Seleccionar el modulo de grupos.



Figura 23. Selección del grupo a consultar.



Figura 24. Despliegue de información del grupo.

4.4.1.4. Módulo de salones.

En éste módulo se presenta al usuario la información del salón consultado. Su utilidad radica en que le permite identificar si el salón que desea reservar cuenta con el suficiente número de computadoras, sistema operativo, modelo y programas instalados. La figura 25 muestra una pantalla de consulta de un salón.



Figura 25. Despliegue de información del salón consultado.

4.4.1.5. Información personal:

Este módulo muestra los datos del usuario, su utilidad radica en que el usuario pueda cambiar su contraseña, así como verificar información de índole personal tal como: nombre, correo electrónico y número de empleado, se muestra en la figura 26.

22159 Annelisse Margarita

[Inicio](#)
[Cerrar sesión](#)

Información personal

Matrícula: 22159
Nombre: Annelisse Margarita
Apellido paterno: Crabtree
Apellido materno: Garcia
Email: crabtree@uabc.edu.mx
Sexo: F
Ocupación: Maestro
Contraseña: [Cambiar](#)

Figura 26. Despliegue de información del empleado.

4.4.1.6. Menú principal para el administrador.

Éste módulo requiere el ingreso de datos del administrador de usuario y contraseña, la página inicial se muestra en la figura 27.



Figura 27. Menú principal del sistema para el administrador.

El menú permite al administrador realizar diversas operaciones, tales como:

Altas: Esta opción permite dar de alta en el sistema nuevos registros: usuarios, administradores, materias, grupos, etc.

Bajas: Mediante esta opción se eliminan del sistema a registros obsoletos: usuarios, administradores, materias, grupos, etc.

Consultas: Esta opción le permite consultar todos los registros existentes en el sistema, la forma de consulta es por listas.

Modificaciones: A través de esta opción no solo se hacen consultas, sino también le permite modificar los registros existentes en el sistema.

Reservaciones: Esta opción le permite reservar los salones existentes en el Laboratorio de Computo, la forma de reservar es por día.

Otros: Se desglosan diferentes actividades del sistema: reportes, asignaciones, cerrar sesión, etc.

F) Tipos de Reportes

Se puede obtener una serie de reportes para documentar y analizar el uso de los espacios. Lo cual sirve como instrumento para la toma de decisiones de rehacer horarios o manejar otras actividades. Los tipos de reportes que genera el sistema son:

- 1) Clases por maestro.
- 2) Clases por grupo.
- 3) Clases por carrera.
- 4) Clases por materia.
- 5) Inventario de software por salón.
- 6) Horas libres.
- 7) Porcentaje de uso de laboratorios.

Uno de los reportes más solicitados y por lo tanto importantes es de **porcentaje de uso** de laboratorios debido a que la información extraída de los datos se utiliza para realizar mantenimiento preventivo y para observar si la administración del mismo ha sido eficiente. En la tabla 1 se muestra un resumen del reporte de uso de un salón.

Tabla 1. Porcentaje de uso mensual de un salón.

Salón 101 Horas de reservación diarias disponibles 15 Horas de reservación fija diarias 8.8						
Mes	Total de días hábiles del mes	Horas de reservación disponibles en el mes	Horas reservadas por el maestro en el mes	Horas fijas reservadas en el mes	Total de horas reservadas en el mes	Capacidad de reservación utilizada
Febrero	17	255	16	149	165.6	64.90%
Marzo	21	315	59	184.8	243.8	77.40%

4.5. Pruebas estáticas

En estas pruebas se especifica cómo se van a realizar las pruebas de inspección, quiénes son las personas encargadas de la inspección del código y el formato con el que se lleva un control de las pruebas realizadas. El formato está diseñado para registrar los datos relevantes durante la revisión, éste formato se muestra en la figura 28.

Probado por :		
Tipo de Prueba:		Estática
Numero de Caso de Prueba:		
Nombre del Caso de Prueba:		
Descripción del Caso de Prueba:		.
Objetos a probar:		
1		
Especificaciones		
Entrada		Salida
Pasos Procedimentales		
1		
2		
3		
4		
5		
6		

Figura 28. Formato para la aplicación de pruebas.

Fuente: Elaboración Propia.

Capítulo V Resultados.

Un punto importante para cualquier sistema de información es la forma de acceder a los datos y lo fácil de su explotación para obtener estadísticas de análisis. El objeto de lo anterior es potenciar decisiones informadas y acciones de acuerdo a datos reales. En este desarrollo uno de los objetivos principales y justificación para su creación fue el poder explotar la información.

De acuerdo a lo planteado en el objetivo b) Obtención de información estadística por carrea, con respecto a la frecuencia de uso del equipo de cómputo. Se realizó el reporte que se muestra en la figura 29, donde se visualiza de forma sencilla el uso por horas de los laboratorios en reservación permanente.

Las reservaciones permanentes las concede el administrador si en la unidad de aprendizaje correspondiente se tienen registradas horas de laboratorio "obligatorias".

LI	96
LAE	20
LNI	0
LC	5
TC	39
TOTAL RESERVACIONES PERMANENTES POR CARRERA	160

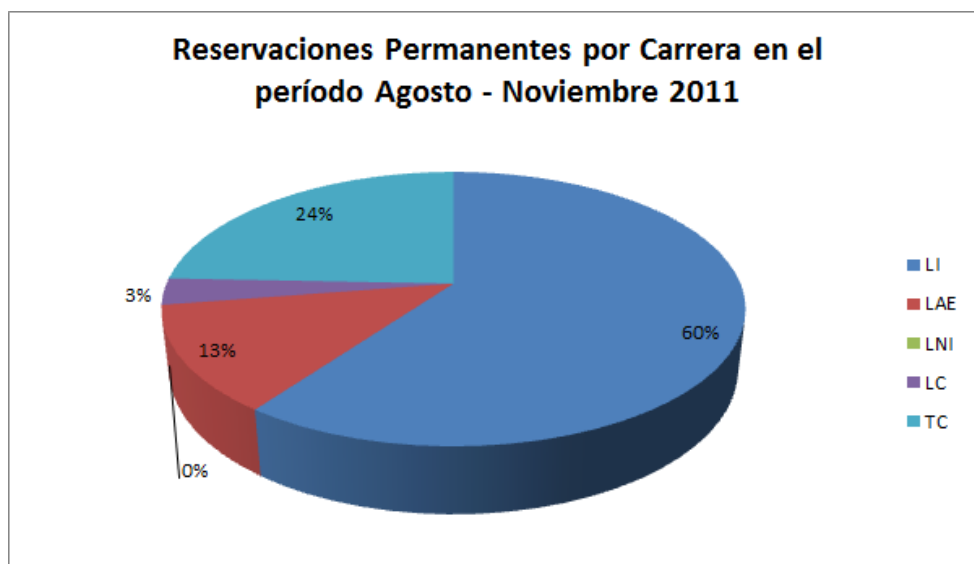


Figura 29. Reporte de reservaciones permanentes por carrera.
Fuente: Elaboración Propia.

Complementado el objetivo b) faltaría las reservaciones que hacen los maestros en forma libre, esto es con el reporte que se muestra en la figura 30 donde se visualiza de forma sencilla el uso por horas de los laboratorios en reservación por los maestros.

RESERVACIONES POR MAESTROS DE TODAS LAS	
CARRERAS	TOTAL 937
LI	203
LAE	354
LNI	60
LC	172
TC	148

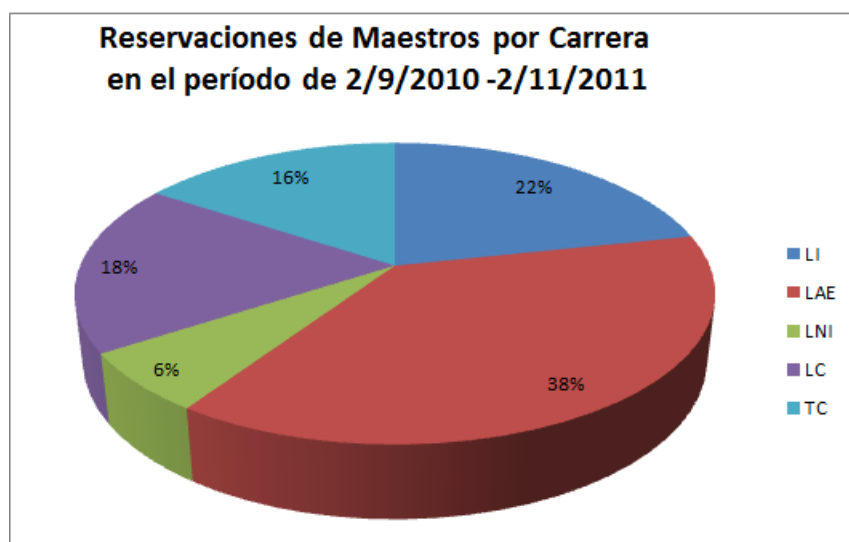


Figura 30. Reporte de reservaciones de maestros por carrera.

Fuente: Elaboración Propia.

Con los siguientes reportes que mostramos en las figuras de la 31 a la 35 se cubre el objetivo a) Implementar estrategias que nos permitan identificar los espacios muertos de los laboratorios estos reportes muestran el uso por carrera por salón y los espacios sin uso de los mismos con esta información ya la dirección o los coordinadores de carrera pueden promover el uso para otros maestros de la facultad.

RESERVACIONES POR CARRERA DEL SALON 101	
Carrera	Total
LAE	80
LI	55
LNI	5
LC	27
TC	51



Figura 31. Reporte de reservaciones salón 101 por carrera.
Fuente: Elaboración Propia.

RESERVACIONES POR CARRERA DEL SALON 102	
Carrera	Total
AE	54
LI	29
LNI	13
LC	28
TC	24



Figura 32. Reporte de reservaciones salón 102 por carrera.

Fuente: Elaboración Propia.

RESERVACIONES POR CARRERA DEL SALON 104	
Carrera	Total
LAE	73
LI	77
LNI	15
LC	50
TC	43



Figura 33. Reporte de reservaciones salón 104 por carrera.

Fuente: Elaboración Propia.

RESERVACIONES POR CARRERA DEL SALON 105	
Carrera	Total
LAE	99
LI	78
LNI	17
LC	25
TC	39

Porcentaje de Reservaciones por Carrera del Salón 105

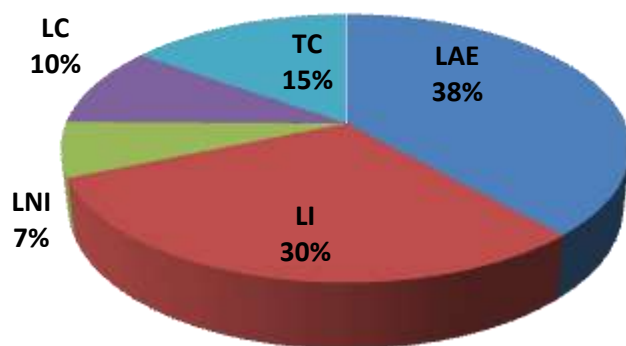


Figura 34. Reporte de reservaciones salón 105 por carrera.

Fuente: Elaboración Propia.

RESERVACIONES POR CARRERA DEL SALON 201	
Carrera	Total
LAE	58
LI	37
LNI	10
LC	45
TC	30



Figura 35. Reporte de reservaciones salón 201 por carrera.

Fuente: Elaboración Propia.

Capítulo VI Conclusiones

Conclusiones

Un centro de cómputo dentro de una organización es un elemento clave para la toma de decisiones y el hecho de contar con una herramienta de software para su administración lo hace más eficiente en sus operaciones diarias, evitando conflictos de disponibilidad, con documentación estadística de uso y sirve como base para agilizar el mantenimiento preventivo de los equipos.

El uso de la arquitectura cliente/servidor es el paradigma de servicios de información más usado porque ofrece la flexibilidad de comunicarse con los usuarios vía internet y permite al administrador descargar la información desde cualquier lugar simplificando en general su trabajo.

La gestión de un centro de cómputo no es una tarea sencilla pero se puede apoyar en las tecnologías de la información para optimizarla, en éste caso las herramientas para el desarrollo del software son de libre acceso para instituciones educativas por lo tanto no se requirió invertir dinero.

Un factor importante en este tipo de soluciones es la parte de la administración información, el costo de licenciamiento y mantenimiento del mismo los cuales se cubrieron gracias al gran avance que ha tenido en los últimos años los programas "Open Source" de los cuales utilizamos Mysql, HTML,PHP, JavaScript, sin generar ningún costo para la institución mientras esté utilizando el sistema.

La información se está resguardando en el manejador de base de Datos Mysql el cual, responde a consultas sobre los datos que contiene, y ejecutar transacciones con la información. Además permite consultas (llamadas query) que se formula como una expresión lógica sobre la información y hace relaciones definidas en el esquema de la base de datos, el resultado es la identificación de un subconjunto lógico de la base de datos, es decir, una selección de registros que cumplen con la administración, resguardo y extracción de la información.

Quizá falte que completes por lo menos una página para darle formalidad al documento, habla sobre las bondades del uso de los recursos de una base de datos mediante "queries" y del avance gracias a programas "open source". Etc.

Anteriormente no se contaba con los mecanismos adecuados que generarán la información, en relación al número de usuarios que se quedaban sin acceso a las salas de cómputo y al uso del equipo. Gracias a la implementación de este software ahora se cuenta con registros que permiten conocer la capacidad utilizada de dicho centro.

Índice de Siglas

FCA (Facultad de Contaduría y Administración)

CACECA (Consejo de Acreditación en la Enseñanza de la Contaduría y Administración A.C.)

IES (Instituciones de Educación Superiores)

Bibliografía

- Ángel Cobo, P. G. (2005). *PHP y MYSQL Tecnologías para el desarrollo de aplicaciones web*. España: Díaz Santos.
- Armand St-Pierre, W. S. (2005). *Redes Locales e Internet*. Trillas.
- CHACÓN, J. C. (2006). *APLICACIÓN DE LA METODOLOGÍA RUP PARA EL*. Guatemala: UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.
- Cohen Karen, D., & Asín Lares, E. (2009). *Tecnologías de Información en los negocios*. Mexico: MCGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V.
- Jiménez, R. H. (2006). *Administración de la función informática: una nueva profesión*.
- Kendall, K. E. (2005). *Analisis y diseño de sistemas*. Mexico.: Pearson.
- Kenneth C. Laudon, J. P. (2008). *Sistemas de Infomacion Gerencial*. Mexico: Pearson Educación de Mexico, S.A. de C.V.
- OZ, E. (2008). *Administracion de los Sisteas de información 5ta edición*. Thomson.
- Parsons, J. J., & Dan, O. (2008). *Conceptos de computación: nuevas perspectivas*. Mexico, DF: C Learninengageg Editores, S.A.
- Ricardo, C. (2009). *Base de Datos*. Mexico: Mc GrawHill.
- Roberto Hernández Sampieri, C. F.-C. (n.d.). *Metodología de la Investigación*. México: McGraw Hill.
- Roger S. Pressman, P. (2010). *Ingeniería de Software un enfoque práctico*. Mexico: McGraw-Hill.
- Sommerville, I. (2006). *Ingeniería del Software* . Madrid España: Pearson Educación S.A.

Cibergrafía

[1]http://www.anuies.mx/servicios/d_estrategicos/documentos_estrategicos/21/4/26.html.

Consultado el 29/08/2010.

[2]www.uelbosque.edu.co/files/.../Pdf/Normaslaboratorios2010_I_V8_1.pdf

Consultado el 29/08/2010

[3] <http://www.mitecnologico.com/Main/AdministracionCentroDeComputo>

Consultado el 21/03/2011

[4]<ftp://ftp.itmerida.mx/Mario/Gestion%20de%20Proyectos%20de%20Software/Documentacion%20RUP/Tesis%20Aplicacion%20RUP.pdf>

Consultado el 23/04/2011

[5] <http://www.masadelante.com/faqs/sistema-operativo>

Consultado el 23/04/2011

[6] http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen3/ciencia3/149/htm/sec_8.htm

Consultado el 24/04/2011

[7] http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos

Consultado el 24/04/2011

[8] <http://www.misrespuestas.com/que-es-la-comunicacion.html>

Consultado el 24/04/2011

[9] http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen3/ciencia3/149/htm/sec_8.htm

Consultado el 05/11/2013

[10] http://es.wikipedia.org/wiki/Base_de_datos

Consultado el 07/11/2012

[12] http://www.danaconnect.com/wiki/index.php/Base_de_datos_web

Consultado el 07/11/2012

[13] http://axitia.com/html/administracion_del_conocimient.html

[14] <http://www.cavsi.com/preguntasrespuestas/que-es-procesamiento-analitico-en-liea-olap/>

[15] <http://es.kioskea.net/contents/bdd/bddintro.php3>

Consultado el 08/11/2012

[16] http://es.wikipedia.org/wiki/Caso_de_uso