



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

“PROCESO ESTADISTICO PARA EL INVENTARIO
DE LA FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN”

Tesis Profesional
Que para obtener el título de:
Licenciado en Ciencias de la Computación

PRESENTA:
Roberto Antonio Velázquez Tlaseca

ASESOR:
M.C.C Eugenia Érica Vera Cervantes

Puebla, Puebla

Marzo 2009

CAPITULO 1 – MARCO TEORICO

1.1 INTRODUCCIÓN

Con los avances que se han tenido en la computación y debido a que en la actualidad el manejo de información de manera segura es de vital importancia, se han desarrollado sistemas de Bases de Datos que permiten al usuario manipular información eficientemente y sin tantos riesgos. Las exigencias de los usuarios respecto a sistemas de información más flexibles han obligado a dedicar una mayor atención a los datos y a su estructuración, buscándose una gestión más racional de la información, por lo cual ha pasado a ser considerada como un recurso fundamental de la organización.

En los últimos años, las bases de datos han experimentado profundos cambios y no son ya, como ocurría hace algunos años, competencia exclusiva de grandes instalaciones con sistemas de información que gestionen millones de registros. Por el contrario, las bases de datos se han extendido al grado de lograr cubrir las expectativas que los usuarios convencionales y expertos reclaman de ellas: flexibilidad, sencillez y eficiencia.

1.2 INGENIERÍA DE SOFTWARE

1.2.1 Introducción a la Ingeniería de Software

“La ingeniería del software es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente en máquinas reales” [6]. La anterior definición no dice mucho sobre los aspectos técnicos de la calidad del software; e incluso omite la mención de la importancia de mediciones y métricas. Tampoco expresa la importancia de un proceso avanzado. Y sin embargo, la definición de Bauer nos proporciona una línea base. ¿Cuáles son los “principios robustos de la ingeniería” aplicable al desarrollo de software de computadoras? ¿Cómo construiremos el software “económicamente” para que sea fiable? ¿Qué se necesita para crear programas de computadora que funcionen eficientemente no en una máquina sino en diferentes “máquinas reales”? Éstas son preguntas que siguen siendo un reto para la ingeniería de software[6].

1.2.2 Proceso, métodos y herramientas de la Ingeniería de Software

La ingeniería del software es una tecnología multicapa que debe descansar sobre un trabajo de organización de calidad. Los cimientos que son la base de la ingeniería de software están orientados hacia la calidad.

El fundamento de la ingeniería de software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería de software. Las áreas claves del proceso forman la base del control de gestión del proyecto del software y establecen el contexto en el que se aplican los métodos técnicos, se producen resultados del trabajo (modelado, documentos, datos, informes, formularios, etc.).

Los métodos de la ingeniería de software indican técnicamente como construir software. Los métodos son: análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Las herramientas de la ingeniería de software proporcionan un soporte automático o semi-automático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado *Ingeniería del software asistido/a por computadora (computer-aided software engineering CASE)* [7].

1.2.3 Características de calidad de un producto de programación

Entre los elementos que inciden en la calidad del software, se tienen:

- Transportable; Es decir debe dar facilidades
- Útil; Es decir que satisfaga las necesidades de la empresa.
- Confiable; Capacidad de un programa para desempeñar una función requerida bajo ciertas condiciones durante un tiempo específico.
- Fácil de probar y mantener; Es decir debe estar bien documentado y ser legible para facilitar la etapa de mantenimiento.
- Eficiente; El sistema debe aprovechar los recursos de hardware y software de la forma más eficiente posible, evitando derrochar tiempo de procesador, memoria, disco, etc.
- Costeable; Un producto de programación debe desempeñar en su empleo diario una tarea específica usando menos tiempo y menos recursos humanos o industriales que los que requiera antes de tenerlo [7].

1.2.4 El ciclo de vida del software

El trabajo de ingeniería de software puede dividirse en tres fases genéricas.

La fase de definición la cual se centra sobre el qué.

- Que información ha de ser procesada.
- Que interfaces van a ser establecidas.
- Qué restricciones de diseño existen.
- Qué función y rendimiento se desea.
- Qué comportamiento del sistema se espera.

- Qué criterios de validación se necesitan para definir un sistema correcto.
- Tareas principales de la fase de definición
- Ingeniería de sistemas o de información.
- Planificación del proyecto del software.
- Análisis de los requisitos.

La fase del desarrollo se centra en el cómo.

- Cómo han de diseñarse las estructuras de datos.
- Cómo ha de implementarse la función como una arquitectura del software.
- Cómo han de implementarse detalles procedimentales.
- Cómo han de caracterizarse las interfaces.
- Cómo ha de traducirse el diseño en un lenguaje de programación.
- Cómo ha de realizarse la prueba.

Tareas principales de la fase de desarrollo:

- Diseño del software.
- Generación del código.
- Prueba del software.

En la ingeniería de software existen varios paradigmas con los que se puede atacar un proyecto en particular, entre los más importantes se tienen los siguientes: de prototipos, de cascada, de espiral.

Para la creación de este proyecto es muy importante seguir una ingeniería de software ya que con esto se garantiza una mejor planeación y calidad del software [7].

1.3 BASES DE DATOS

1.3.1 Conceptos básicos sobre bases de datos

Comenzaremos diciendo que una base de datos es un **conjunto, colección o depósito de datos** almacenados en un soporte informático no volátil. Los datos están interrelacionados y estructurados de acuerdo con un modelo capaz de recoger el máximo contenido semántico. Dada la relevancia que tienen en el mundo real las interrelaciones entre los datos, es imprescindible que la base de datos sea capaz de almacenar estas interrelaciones. En el mundo real existen, además restricciones semánticas, a las que se está concediendo una importancia creciente y que, en los sistemas actuales, tienden a almacenarse junto con los datos, al igual que ocurre con las interrelaciones [2].

La **redundancia** de los datos debe ser controlada, de forma que no existan duplicaciones perjudiciales ni innecesarias, y que las redundancias físicas, convenientes

muchas veces a fin de responder a objetivos de eficiencia, sean tratadas por el mismo sistema, de modo que no puedan producirse inconsistencias. Esto se podría resumir diciendo que en las bases de datos no debe existir redundancia lógica, aunque si se admite cierta redundancia física por motivos de eficiencia. Por tanto, un dato se actualizará lógicamente por el usuario de forma única, y el sistema se preocupará de cambiar físicamente todos aquellos campos en los que el dato estuviese repetido en caso de existir redundancia física.

Una de las principales características de las bases de datos es la **independencia**, tanto física como lógica, entre datos y tratamientos. Esta independencia, objetivo fundamental de las bases de datos, es una característica esencial que distingue las bases de datos de los archivos y que ha tenido una enorme influencia en la arquitectura de los sistemas de administración de bases de datos (data base management system - DBMS).

La definición del conjunto de datos contenidos en la base (lo que se denomina estructura de la base de datos) debe ser única y estar integrada con los mismos datos. En las bases de datos, la descripción, y en algunos casos también la definición y documentación completas (metadatos), se almacena junto con los datos de modo que estos están auto documentados, y cualquier cambio que se produzca en dicha documentación se ha de reflejar y quedar recogido en el sistema.

La actualización y recuperación de datos se debe realizar mediante procesos bien determinados, incluidos en el DBMS, así mismo, ha de proporcionar instrumentos que faciliten el mantenimiento de la seguridad de los datos.

Visto lo anterior, podemos definir la base de datos como:

“Colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada. Los datos que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos, y su definición (estructura de la base de datos) única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos” [6].

Las bases de datos pretenden servir al conjunto de la organización, manejando los datos como otro recurso que viene a añadirse a los ya tradicionales. Por tanto, las bases de datos han de atender a múltiples usuarios y a diferentes aplicaciones.

Cabe mencionar que las bases de datos contienen tres capas de abstracción, cada una de ellas separada en cierta medida de cambios en las capas inferiores. En el fondo de la jerarquía está la capa física, que trata con estructuras de almacenamiento. Esta capa contiene estructuras de datos de bajo nivel (por ejemplo, archivos, bloques de control, aceleradores de búsqueda), y aloja las funciones del hardware. La capa física contiene servicios de soporte para la capa conceptual, que proyecta la ilusión de tablas y objetos, a partir de los cuales el usuario moldea el entorno de aplicación. El DBMS puede proteger la capa conceptual de ciertos cambios a la capa física, característica que se conoce como independencia física de datos. La capa superior representa vistas personalizadas de la capa

conceptual adecuadas para diferentes usuarios. El DBMS puede proteger la capa superior contra cambios limitados al esquema conceptual, función que se conoce como independencia lógica de datos.

Por último, el Administrador de una Base de Datos es una persona responsable de los esquemas conceptual y físico, respaldo periódico, recuperación según las necesidades y temas de operación y seguridad. El Administrador es la principal persona de contacto para usuarios que tienen sugerencias para modificar la estructura de la base de datos.

1.3.2 Ventajas e inconvenientes de utilizar Bases de Datos

El utilizar sistemas de bases de datos nos proporciona ciertas ventajas, como son:

1. **Compartir los datos:** Compartir no solo significa que las aplicaciones existentes puedan compartir la información de la base de datos, sino también que sea posible desarrollar nuevas aplicaciones para operar sobre los mismos datos.
2. **Nos permite disminuir la redundancia:** En los sistemas sin bases de datos cada aplicación tiene sus propios archivos privados. Esto puede provocar considerable redundancia en los datos almacenados, con el consecuente desperdicio de espacio de almacenamiento.
3. **Se puede evitar la inconsistencia:** Esta se da cuando un dato está representado por dos entradas distintas, con lo cual la base de datos puede proporcionar información incorrecta o contradictoria a sus usuarios. Este tipo de situaciones se evitan por medio del proceso de propagación de actualizaciones, el cual incluye las operaciones de inserción, eliminación y modificación.
4. **Es posible brindar un manejo de transacciones:** Una transacción es una unidad de trabajo lógica, que por lo regular comprende varias operaciones de la base de datos (operaciones de actualización).
5. **Es posible mantener la integridad:** La integridad de una base de datos consiste en asegurar que la información almacenada sea correcta. Un ejemplo de falta de integridad podría ser la inconsistencia entre dos entradas que supuestamente representan al mismo hecho.
6. **Se pueden aplicar restricciones de seguridad:** Se puede asegurar que el acceso a la base de datos sea sólo a través de los canales apropiados, y se pueden definir las verificaciones de seguridad por realizar cuando se intente acceder a información delicada.
7. **Es posible equilibrar requerimientos en conflicto:** Al conocer los requerimientos generales de una empresa, se puede estructurar el sistema con miras a proporcionar un servicio general óptimo para la empresa. Por ejemplo, se puede escoger una forma de representación de los datos almacenados con la cual las aplicaciones más

importantes puedan tener un acceso rápido, aunque el funcionamiento de algunas otras aplicaciones sufra menoscabo.

8. Es posible cumplir los estándares: Con el control central de la base de datos, es conveniente estandarizar la representación de datos, en particular como un auxiliar para el intercambio de datos o para el movimiento de datos entre sistemas. Así mismo, los estándares en la asignación de nombres y en la documentación de los datos también son muy convenientes como una ayuda para compartir y entender los datos.
9. Normalización: Al tener un control centralizado de la base de datos, se puede garantizar la observancia de todas las normas aplicables para la representación de los datos. La normalización de formatos de los datos almacenados es deseable sobre todo como apoyo para el intercambio de información entre sistemas. Las normas para nombrar y documentar los datos son muy convenientes como ayuda para el compartimiento y comprensibilidad de la información [1].

Las bases de datos no sólo presentan ventajas, sino que también tienen posibles inconvenientes, como son:

1. Instalación costosa: La implantación de un sistema de bases de datos puede llevar consigo un costo elevado, tanto en equipo físico como en el lógico, además del mismo coste de adquisición y mantenimiento.
2. Personal especializado: Los conocimientos, que resultan imprescindibles para una utilización correcta y eficaz y sobre todo para el diseño y administración de las bases de datos, implican una necesidad de personal especializado.
3. Implantación larga y difícil: Al implantar una base de datos, van apareciendo a lo largo de su desarrollo dificultades, que llevan en general a que se superen los plazos inicialmente previstos.
4. Falta de rentabilidad a corto plazo.
5. Desfase entre teoría y práctica: Al existir un considerable avance de la teoría en relación con la práctica, en muchas ocasiones los usuarios, especialmente los directivos, se engañan respecto a las prestaciones reales que pueden proporcionarles los DBMS actuales, creyendo que constituyen ya una realidad ciertos aspectos que todavía son solo teóricos [2].

1.3.3 Que son los Sistemas Manejadores de Bases de Datos (DBMS)

Comenzaremos por dar un ejemplo sencillo, para hacer notar porque es indispensable un sistema de gestión de base de datos (DBMS Data Base Management System). Antes de que existieran los DMBS, digamos los bancos utilizaban un programa diferente para cada una de las siguientes tareas:

- Hacer cargos o abonos a una cuenta.
- Dar de alta una nueva cuenta.
- Obtener el saldo de una cuenta.
- Generar estados de cuenta mensuales.

El hecho de que diferentes programas manejen la misma información genera los siguientes problemas:

- Redundancia e inconsistencia de datos.
- Dificultad para tener acceso a los datos.
- Anomalías de acceso concurrente.
- Problemas de integridad.

Un objetivo importante del DBMS es proporcionar a sus usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Puesto que muchos usuarios de sistemas de Base de Datos no tienen experiencia en computadoras, se les esconde la complejidad a través de diversos niveles de abstracción.

Los niveles de abstracción son los siguientes:

- *Nivel Físico*: Describe como se almacena realmente la información en los dispositivos de almacenamiento (HD, cinta, etc.)
- *Nivel Conceptual*: Describe que datos son almacenados en la Base de Datos y como es la relación entre ellos (nombre, teléfono, dirección, etc.)
- *Nivel visión*: Es el nivel más alto de abstracción y describe solamente parte de la Base de Datos. A muchos usuarios no les interesa toda la información, porque solamente trabajan con una parte de ella, comúnmente son llamadas vistas.

Un sistema de gestión de base de datos (DBMS Data Base Management System) consiste en la colección de datos interrelacionados y un conjunto de programas para acceder a los datos.

El objetivo primordial de un DBMS es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información en la base de datos.

Los DBMS deben mantener la seguridad de la información almacenada, pese a caídas del sistema o intento de acceso no autorizado. Si los datos van a ser compartidos por varios usuarios, el sistema debe evitar los problemas de concurrencia.

1.3.4 Tareas y funciones generales del DBMS

- Las solicitudes hechas por el usuario, son interpretadas y analizadas por el DBMS [3].
- Soporta una gran variedad de operaciones sobre la base de datos, así también tiene capacidad de deshacer algunos comando no deseados.
- Deben existir componentes procesadores para los DDL y los DML [4].
- Obviamente debe proporcionar una seguridad y una integridad de los datos, así como un diccionario de datos.
- Contar con algoritmos óptimos para agilizar las sentencias dadas. Y un plan de ejecución eficiente.

1.3.5 El modelo de datos y sus cualidades

Para que nuestro modelo de datos se pueda considerar un buen modelo [5] debe cumplir con lo siguiente:

- Expresividad
- Simplicidad
- Minimalidad
- Formalidad

La *expresividad* depende en gran parte de los conceptos con que cuente, así como la elección de cada una de ellas.

Así tenemos que un modelo de datos puede ser muy expresivo, pero depende de la variedad de conceptos con que se cuente, aún cuando no siempre la cantidad lo dice todo.

La *simplicidad* es que tan entendible puede ser un esquema una vez que se haya generado, el cual debe ser de fácil entendimiento para el analista como para el cliente o usuario.

La *minimalidad* nos proporciona unicidad de conceptos, es decir que cada concepto no se pueda derivar de los demás.

La *formalidad* tiene mucho que ver con la cualidad de minimalidad, si un concepto es único, por consecuencia tendrá una interpretación y definición única.

Por otro lado un modelo de datos debe tener las siguientes propiedades:

- Complejión gráfica
- Facilidad de Lectura

La complejidad gráfica es la propiedad de que cada concepto perteneciente al modelo, cuente con una representación gráfica o símbolo. La facilidad de lectura debe cumplir la propiedad de que cada concepto del modelo tenga asignada una representación gráfica única que lo haga distinguible del resto de los conceptos. De esta forma se evita la confusión entre los conceptos.

1.3.6 El Modelo Entidad - Relación (E-R)

Es una técnica de diseño de bases de datos gráfica, que incorpora información relativa a los datos y la relación existente entre ellos, para poder así plasmar una visión del mundo real sobre un soporte informático. Sus características fundamentales son:

- Reflejan tan sólo la existencia de los datos sin expresar lo que se hace con ellos.
- Es independiente de las bases de datos y de los sistemas operativos.
- Incluye todos los datos que se estudian sin tener en cuenta las aplicaciones que se van a tratar.
- Las entidades se representan como rectángulos, los atributos como elipses y las relaciones como rombos [5].

1.3.7 Cualidades del modelo E-R

El modelo de datos Entidad-Relación es un modelo rico en conceptos, por lo cual tiende a ser *expresivo* y por consiguiente bastante entendible tanto para el analista de sistemas como para el cliente; así por ejemplo podemos representar las entidades con rectángulos con esquinas redondeadas sabiendo que cada entidad es una tabla en nuestra base de datos, aunque no todas las entidades tienen una representación en la base de datos, otro claro ejemplo son los atributos, que pueden ser representados como opcionales u obligatorios, para el primer caso basta con representarlos con un círculo sin rellenar y para los obligatorios solo rellenamos el círculo, para el caso de las llaves primarias se pueden representar con un gato (#) Para las relaciones entre entidades podemos representarlas con líneas punteadas o continuas, indicando también en los extremos de cada línea si es de muchos a uno o uno a uno, de muchos a muchos y cuando se trata de estos últimos hay que agregar una entidad intermedia para que la relación quede de muchos a uno, en los capítulos posteriores explicaremos esto con mayor claridad; lo dicho anteriormente es para dejar en claro lo entendible que es un diagrama E-R.

La *minimalidad* es cubierta, pues ningún concepto del modelo puede derivarse por la combinación de otros conceptos del mismo modelo, con ello queda también cubierto el concepto de *formalidad*, pues cada concepto es único.

Por otra parte cumple también con las propiedades de *representación gráfica*, puesto que cada uno de los conceptos tiene asignado su propio símbolo que lo identifica como único en el modelo, y por consiguiente es de fácil lectura pues, una vez

comprendidos sus conceptos y su representación gráfica la lectura del esquema resulta sencilla y fácil [5].

1.4 PAGINAS WEB DINAMICAS

1.4.1 Introducción a ASP

ASP, abreviatura de Active Server Pages, es una plataforma creada por Microsoft, destinada a servir como soporte para la creación de páginas web de contenido dinámico.

Sus principales características son las siguientes:

- *Se ejecuta en el servidor:* No es el navegador quien ejecuta código ASP, sino el servidor Web, quien mandará el resultado de la ejecución al navegador.
- *Se programa en lenguaje script:* No es un lenguaje de programación propiamente dicho, ya que no sirve para crear programas independientes. En su lugar, los lenguajes script se entremezclan dentro de un documento (por ejemplo una página HTML) para dar funcionalidad en este documento e interactuar con él.

Esto implica una serie de ventajas respecto al HTML, que son las siguientes:

- *Mayor seguridad:* Al ejecutarse en el servidor, el código fuente nunca es enviado al navegador, con lo que al internauta le es imposible obtener el código fuente de nuestras páginas.
- *Mayor funcionalidad:* Al ejecutarse en el servidor, podemos realizar cosas imposibles de realizar en el cliente, como por ejemplo, guardar datos en una base de datos, compartir datos entre distintos usuarios (por ejemplo, la base de datos para un catalogo de material o equipo), etc.
- *Mayor compatibilidad con los navegadores:* Al poder realizar toda la programación en el servidor, es posible generar páginas que contengan exclusivamente HTML, con lo cual no estamos forzando a que el navegador deba soportar JavaScript o algún otro tipo de programación que se ejecuta en el navegador.
- *Lenguaje más fácil:* ASP se suele programar en Visual Basic Script (VBScript), el cual es un lenguaje casi idéntico a Visual Basic, lenguaje muy popular por su facilidad y velocidad de aprendizaje.
- *Multi-lenguaje:* ASP no es, en realidad, un lenguaje de programación, sino una plataforma de soporte para la programación de distintos lenguajes script en el servidor. Ello implica que, aunque el lenguaje utilizado suele ser VBScript, también podemos usar otros lenguajes como JavaScript, PerlScript, Rexx, Phyton, y muchos otros. Esto hace que si un programador conoce muy bien otro lenguaje script, pueda utilizarlo en lugar de VBScript, aprovechando sus conocimientos y experiencia ya adquiridos, es decir, no está obligado a aprender un lenguaje nuevo [8].

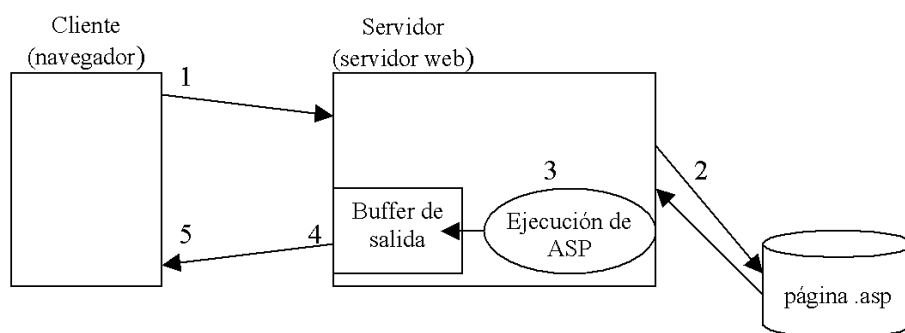
1.4.2 Interacción entre el servidor y el navegador web cuando se utiliza ASP.

ASP no se ejecuta en el navegador, sino en el servidor web. Cuando un internauta hace una petición de una página ASP se producen los siguientes acontecimientos:

1. El navegador localiza al servidor web y le lanza una petición http donde se le solicita una página ASP.
2. El servidor busca esta página asp en su disco (o en su caché) y si no la encuentra genera un error. Si la encuentra la carga en el servidor.
3. El servidor interpreta la página ASP, la cual contiene mezclados código HTML y código script de servidor. Cuando encuentra código HTML lo envía "tal cual" a un buffer de salida. Cuando encuentra código script de servidor, lo ejecuta, y el resultado generado es enviado al mismo buffer de salida.
4. El servidor envía el buffer de salida al navegador, el cual contendrá únicamente código HTML.
5. El navegador recibe el documento Html (aunque la extensión que aparece en la URL sea .asp).

Como se puede apreciar, el paso más importante es el paso 3, donde en caso que fuera una página HTML, simplemente se copiará el fichero del disco al buffer de salida.

El siguiente esquema resume estos pasos:



1.4.3 Otras alternativas

ASP no es la única alternativa de desarrollo web. Existen otras plataformas como JSP (Java Server Pages, de Sun Microsystems), ColdFusion (de Aflaire) o PHP (de libre distribución bajo Open System).

Algunas de las ventajas que ofrece ASP respecto a estas otras alternativas son:

- Su bajo costo (frente a JSP o ColdFusion), pues el servidor web viene incluido con los sistemas operativos de Microsoft.
- La facilidad del lenguaje, pues ASP se suele programar en VBScript, lenguaje mucho más simple que Java o PHP (el cual es similar al C).
- Es, probablemente, la tecnología más usada (sobretudo si lo comparamos con JSP o ColdFusion). Esto hace que sea muy fácil obtener información sobre el tema totalmente gratis, pues hay muchas páginas en internet con manuales, tutoriales, artículos, o código fuente de páginas ASP.
- Existen servidores gratuitos para hospedar nuestras páginas ASP, y los que son de pago (los cuales suelen ofrecer una mayor funcionalidad) no suelen ser caros.
- ASP utiliza componentes Active-X, los cuales amplían las funcionalidades de la plataforma ASP. Se puede encontrar en internet una cantidad innumerable de componentes Active-X que pueden ser usados en nuestras páginas ASP.

1.4.4 Herramientas necesarias

1.4.4.1 El servidor web

Lo primero que necesitamos para ejecutar una página ASP es tener instalado en el servidor un servidor web que soporte ASP. Dado que ASP es una tecnología propietaria de Microsoft, existen básicamente 2 servidores web, los 2 de Microsoft, los cuales soportan ASP: IIS y PWS.

- IIS (Internet Information Server) es la solución profesional. Es decir, es un servidor web que soporta muchos usuarios conectados a la vez, y ofrece un rendimiento muy bueno. IIS viene incluido en los sistemas operativos basados en tecnología NT Server, tal como Windows-NT Server 4.0 o Windows-2000 Server.
- PWS (Personal Web Server) es la solución para el desarrollo. Es decir, es un servidor web con la misma funcionalidad que IIS, pero que soporta un limitado número de usuarios conectados a la vez. Concretamente, suele estar limitado a 10 usuarios a la vez, con lo que sirve a la perfección para desarrollar nuestro sitio, hacer pruebas, etc., pero no sirve, en la mayoría de los casos, para hacer de servidor real de nuestra web, a no ser que ésta tenga pocas visitas. La ventaja de PWS es que éste se incluye en los sistemas operativos personales de Microsoft, como Microsoft Windows-95 y Microsoft Windows-98. Concretamente, el PWS se encuentra en un directorio del CD de instalación llamado Add-ons [8].

1.4.4.2 El navegador

Con el fin de poder probar nuestras páginas ASP, necesitaremos también de un navegador, que bien puede ser Internet Explorer, Netscape Navigator, o cualquier otro que nos permita visualizar páginas HTML.

1.4.4.3 Herramientas para el desarrollo

Una página ASP, no es más que una página HTML que contiene código ASP dentro. Es decir, una página ASP no es más que un fichero de texto llano. Por este motivo, podemos utilizar nuestro editor de texto preferido, como por ejemplo el bloc de notas del Windows, el editor del MS-DOS, o cualquier editor de HTML como por ejemplo Dreamweaver.

Existen también herramientas específicas para la ayuda al desarrollo de las páginas ASP, tal como Microsoft Visual InterDev, incluido en la Suite Microsoft Visual Studio [8].

CAPITULO 2 – ANALISIS Y DISEÑO

2.1 ANALISIS

2.1.1 Definición del problema

A pesar de la amplia aplicación de las bases de datos dentro de todo tipo de sistemas para empresas e instituciones no lucrativas, no es suficiente para quienes tienen un propósito muy en particular para dar solución a un problema.

Este proyecto atiende principalmente la necesidad de una institución educativa para llevar a cabo el control total de inventario de su equipo de cómputo y multimedia ya que es sujeto a préstamo, además de contar con una serie de reportes que permitirán obtener conclusiones sobre la demanda y uso de material en específico.

Debido a que el sistema contará con disponibilidad en la web podrá ser accesado desde cualquier punto remoto de internet y lo que permitirá realizar la consulta para verificar si existe lo que se requiere y si está o no disponible.

2.2 ESPECIFICACION DE REQUERIMIENTOS

2.2.1 Recopilación

Mediante la construcción de prototipos de software basados en entrevistas verbales con el administrador del inventario de la institución se lograron identificar diversas características.

En primera instancia, los perfiles de usuarios del sistema:

- Administrador
- Usuario Estándar.

Se requiere almacenar información sobre:

- Profesores
- Hardware
- Software
- Libros y Revistas

2.2.2 Antecedentes

Inicialmente el control de inventarios se llevaba en hojas de papel con tabulaciones formando tablas de manera que fuera de cierta manera controlado cada artículo que lo conformaba. Con el nacimiento de las computadoras ya se podía al menos evitar cantidades

muy grandes de información escrita o impresa pero aun así las búsquedas de información y actualizaciones del inventario en gran parte seguían siendo manualmente.

Con la creación de bases de datos en la PC, se ha ido mejorando de manera considerable la administración de grandes volúmenes de datos y se han creado infinidad de aplicaciones que las manejan. Pero aunque ya existen muchas soluciones a muchos problemas de esta índole, aun persiste la necesidad de crear aplicaciones dedicadas a las necesidades específicas de cada empresa.

Las aplicaciones que manejan inventarios lo hacen de manera general y eficiente, pero tienen la desventaja que hay que adaptarse a su manera de trabajo para mantenerlo funcionando correctamente, además de que los sistemas están únicamente instalados en un lugar en específico y el hecho de trasladar la información se vuelve una tarea física, es decir trasladar físicamente el equipo con la información.

Por ello, la propuesta de hacer un sistema que funcione en la Web que solo requiere de tener acceso a Internet para poder disponer y utilizar el sistema desde cualquier parte del mundo.

2.2.3 Alcances

Este proyecto tiene como objetivos los siguientes alcances:

1. Generar un sistema Web con bases de datos que realice los siguiente:
 - Captura, Modificación y Eliminación de información acerca de Profesores, Libros y Revistas, Hardware, Software y prestamos de material.
 - Se limitara el manejo de la información dependiendo del usuario que acceda al sistema.
 - Se podrán consultar diversos tipos de informes sobre el comportamiento y movimientos realizados al inventario.
 - Sera un sistema con una interfaz amigable para los usuarios y otorgara toda la información pertinente y disponible del inventario de manera rápida y clasificada, además de permitir realizar la búsqueda de algún material de manera eficaz.
2. Generar el documento de Tesis.

2.2.4 Metodología

Primeramente se usará el modelo incremental de desarrollo del software, es decir, el análisis, diseño, codificación, prueba y mantenimiento del sistema se harán para cada avance o etapa que se construya del sistema.

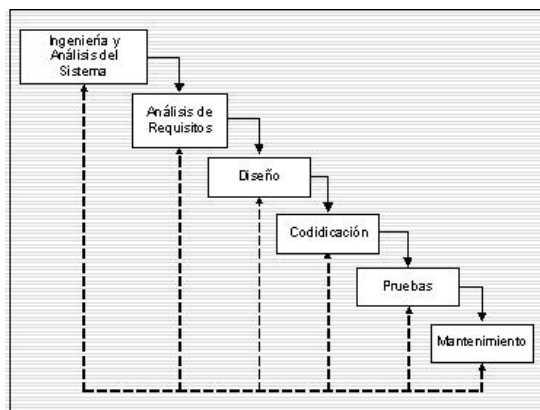
En el análisis se definirá principalmente el planteamiento del problema así como toda la especificación de requerimientos del sistema, funcionales y no funcionales; además se establecerán alcances y restricciones para el sistema.

Para el diseño, apoyados en herramientas de UML, los requerimientos establecidos se definirán a más detalles mediante la representación de casos de uso.

Para la implementación del sistema, basados en los requerimientos funcionales, como se requiere de tecnología de servidor, Internet Información Server será quien cubra ese aspecto, el lenguaje apropiado será ASP utilizando bases de datos en MySQL por su manejabilidad.

Las pruebas de los avances del sistema serán directamente con el encargado del inventario y así dar mantenimiento de acuerdo a las necesidades de las funciones del sistema.

Todo lo anterior lo podemos resumir en la siguiente figura:



2.2.5 Restricciones

1. El sistema solo permite realizar operaciones que afectan a la base de datos a profesores dados de alta en el sistema, si alguno no está, aunque pertenezca a la facultad, no tendrá derecho al servicio de préstamo.
2. Existirán los siguientes permisos para los usuarios:
 - Administrador: Permiso para agregar, modificar y eliminar información, Acceso total a la base de datos, Consulta de Reportes estadísticos.
 - Usuario Estándar: Permiso de Consulta de material e impresión de hoja de solicitud de préstamo.
3. El sistema solo proporcionará información dependiendo del rol del usuario.

2.2.6 Requerimientos Del Sistema

Los objetivos de este proyecto son proporcionar una herramienta útil y de fácil manejo a los usuarios, dando así un servicio de calidad y oportuno para las operaciones diarias del equipo que se llevan a cabo en la Facultad de Ciencias de la Computación de la BUAP.

1. En la institución, el equipo se encuentra en un almacén al cual llegan los profesores a solicitar algún material, ya sea desde un equipo de cómputo hasta un libro o una revista. Se requiere llevar el control de ese almacén para lograr un manejo más eficiente y confiable de la información.
2. Debido a la extensión del inventario se requiere de un catálogo que clasifique el equipo y material con el que se cuenta por categorías, además de un buscador interno para averiguar si se cuenta o no con algún material así como sus características (descripción, ubicación, disponibilidad, etc.).
3. Cada que un profesor necesita de algún equipo en específico debe acudir al almacén, preguntar al encargado si existe y verificar la disponibilidad de lo que necesita, enseguida llenar su correspondiente formato de préstamo. Se requiere entonces que mediante el sistema, parte de este procedimiento se haga desde la página web para mayor comodidad y seguridad.
4. Debido a que el inventario tiene movimientos constantes, se requiere generar reportes estadísticos que ayuden a mejorar este servicio en la institución.

2.2.7 Requerimientos Funcionales

- Control del equipo en el almacén, clasificado y organizado de manera estratégica en la base de datos.
- Programación de las operaciones que ha de realizar un profesor en el sistema.
- Seguimiento del material sobre disponibilidad, ubicación y préstamo.
- Generación de reportes semanales o mensuales sobre el uso del material.
- La confiabilidad de las transacciones se controlara por llaves primarias y foráneas, y las restricciones estarán implementadas en el SGBD.

2.2.8 Requerimientos No Funcionales

- Capacitación a los profesores sobre el uso y la organización del sistema.
- Difundir entre los usuarios algunas recomendaciones sobre el uso responsable del sistema.

2.2.9 Descripción de la Información

a) Casos de uso

CASO DE USO : INGRESO AL SISTEMA	
ACTOR: ADMINISTRADOR	
CURSO NORMAL	ALTERNATIVA
1) El administrador tecleara su password en un apartado exclusivo para él.	En caso de teclear una contraseña inválida, volverá a la página de inicio sin avanzar.
2) El administrador se conectara y tendrá acceso a todas las modificaciones del sistema.	

CASO DE USO: ALTA DE MATERIAL	
ACTOR: ADMINISTRADOR	
CURSO NORMAL	ALTERNATIVA
1) El administrador elegirá la categoría del material en un menú.	
2) Aparecerá el catalogo de material existente en el inventario en la categoría elegida.	
3) El administrador elegirá la opción ALTA dentro de la ventana actual.	
4) Ingresara los datos correspondientes y solicitados por el sistema.	4.1) En caso de no introducir los campos requeridos el sistema manifestara un error e indicara al usuario la razón.
5) EL sistema refrescara la vista para mostrar el nuevo material dado de alta en el catalogo.	

CASO DE USO : BAJA DE MATERIAL	
ACTOR: ADMINISTRADOR	
CURSO NORMAL	ALTERNATIVA
1) El administrador elegirá la categoría del material en un menú.	

2) Aparecerá el catalogo de material existente en el inventario en la categoría elegida.	
3) El administrador elegirá un artículo del catalogo y presionara el botón BAJA dentro de la ventana actual.	
4) EL sistema refrescara la vista para mostrar el material donde ya no aparecerá el previamente dado de baja.	

CASO DE USO : MODIFICACION DE MATERIAL	
ACTOR: ADMINISTRADOR	
CURSO NORMAL	ALTERNATIVA
1) El administrador elegirá la categoría del material en un menú.	
2) Aparecerá el catalogo de material existente en el inventario en la categoría elegida.	
3) El administrador elegirá un artículo de la lista y presionara el botón MODIFICAR dentro de la ventana actual.	
4) Aparecerá un formulario con la información del material lista para hacer correcciones o agregar información.	4.1) En caso de dejar algún campo vacío pero que era requerido el sistema mandara un error para indicar el fallo al administrador
5) EL sistema refrescara la vista para mostrar el material actualizado en el catalogo.	

CASO DE USO: PRESTAR MATERIAL	
ACTOR: ADMINISTRADOR	
CURSO NORMAL	ALTERNATIVA

1) El administrador elegirá la categoría del material en un menú.	
2) Aparecerá el catalogo de material existente en el inventario en la categoría elegida.	
3) El administrador elegirá un artículo del catalogo y presionara el botón PRESTAMO dentro de la ventana actual.	
4) Aparecerá un formato de préstamo auto llenado con la información del ítem elegido.	
5) Se le solicitara a la persona su clave de profesor para identificarlo en el sistema y poder darle el servicio.	5.1) En caso de no ser un profesor dado de alta en el sistema no se le podrá dar el servicio de préstamo de material.
6) El administrador presionara el botón IMPRIMIR HOJA DE PRESTAMO y solicitara la firma del profesor.	
7) Se hará la entrega física del material	

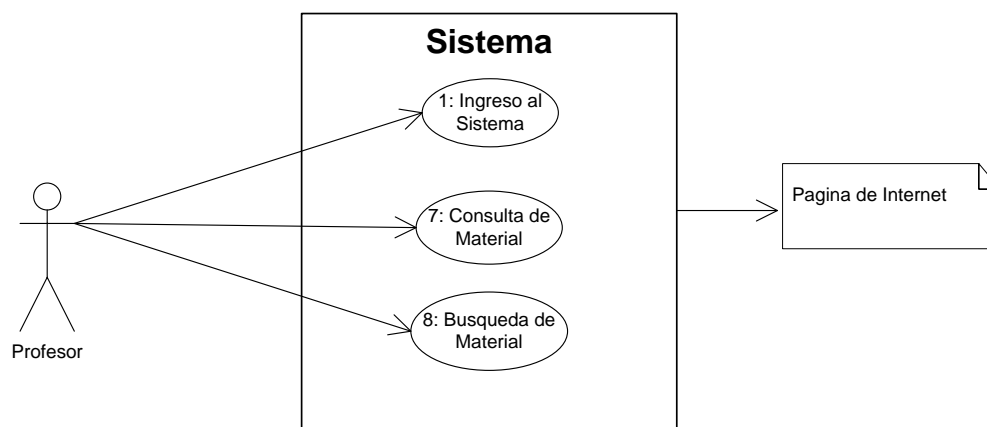
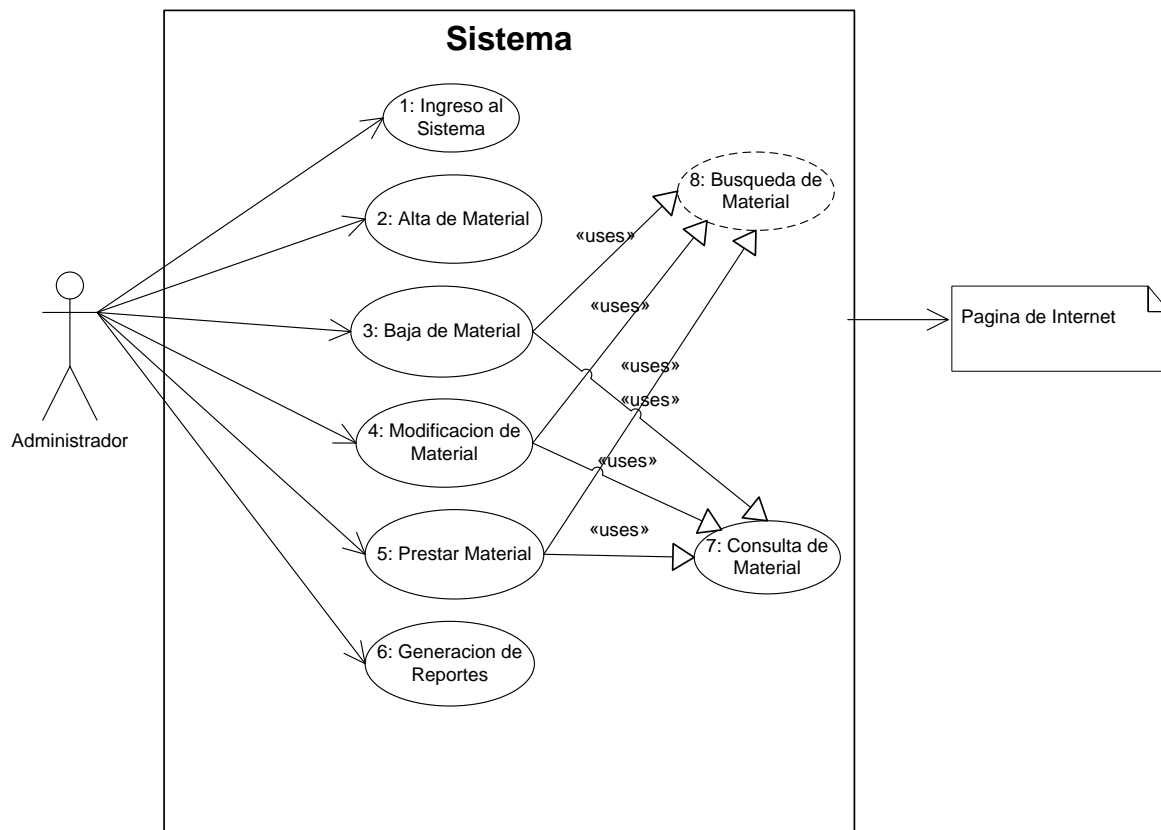
CASO DE USO : CONSULTA DE MATERIAL	
ACTOR: PROFESOR	
CURSO NORMAL	ALTERNATIVA
1) El profesor entrara al sistema mediante una opción dedicada a él.	
2) El profesor elegirá la categoría del material en un menú.	
3) Aparecerá el catalogo de material existente en el inventario en la categoría elegida, incluyendo todos sus detalles, y especialmente su disponibilidad.	

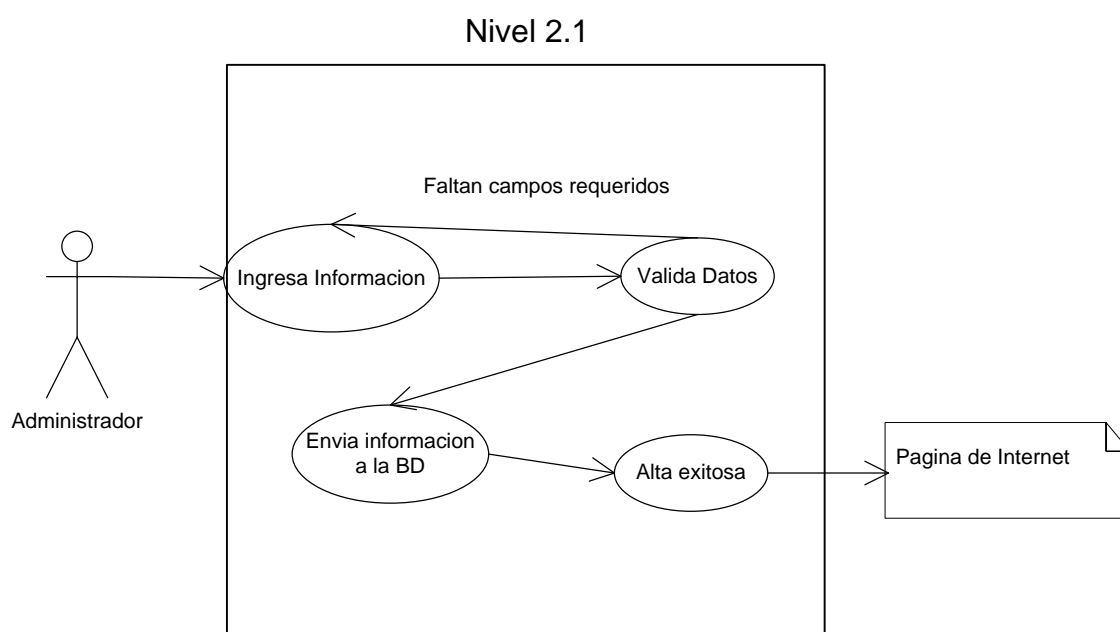
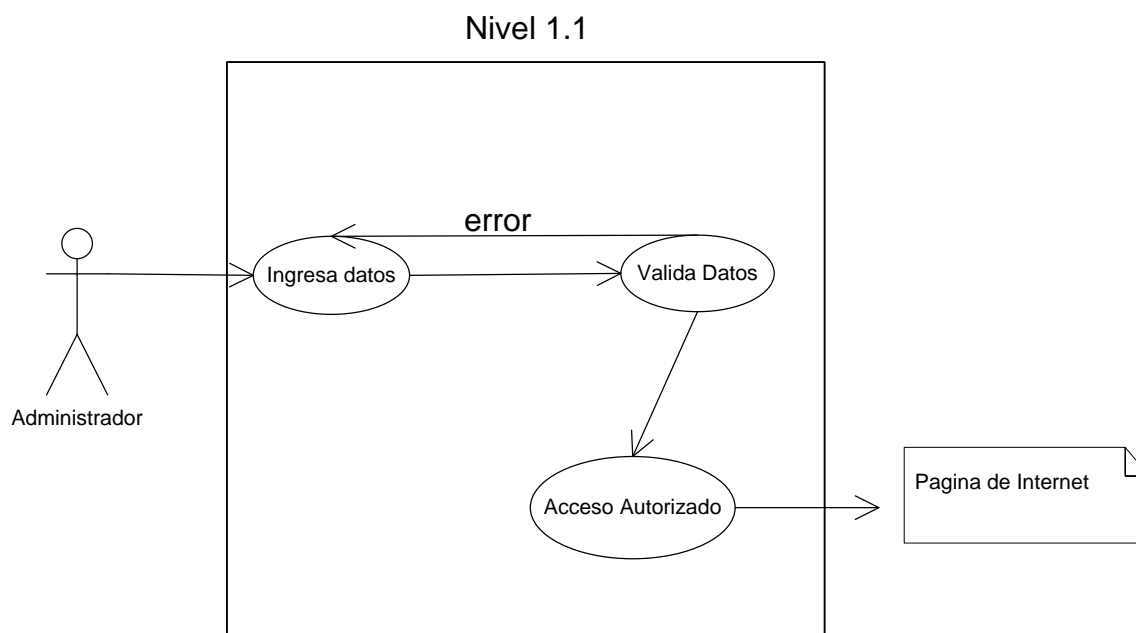
CASO DE USO: BUSQUEDA DE MATERIAL	
ACTOR: ADMINISTRADOR, PROFESOR	
CURSO NORMAL	ALTERNATIVA

1) El usuario entrara al sistema.	
2) Elegirá la opción búsqueda de material.	
3) Deberá elegir una categoría de material y pondrá en el campo de búsqueda la o las palabras clave para identificarlo en la base de datos.	
4) Se desplegara entonces la lista de resultados.	5) En caso de ser administrador se tendrán las opciones de alta, baja, modificación y préstamo disponibles en la hoja de resultados.
5) Si se requiere otra búsqueda se deberá hacer el procedimiento a partir del paso 2.	

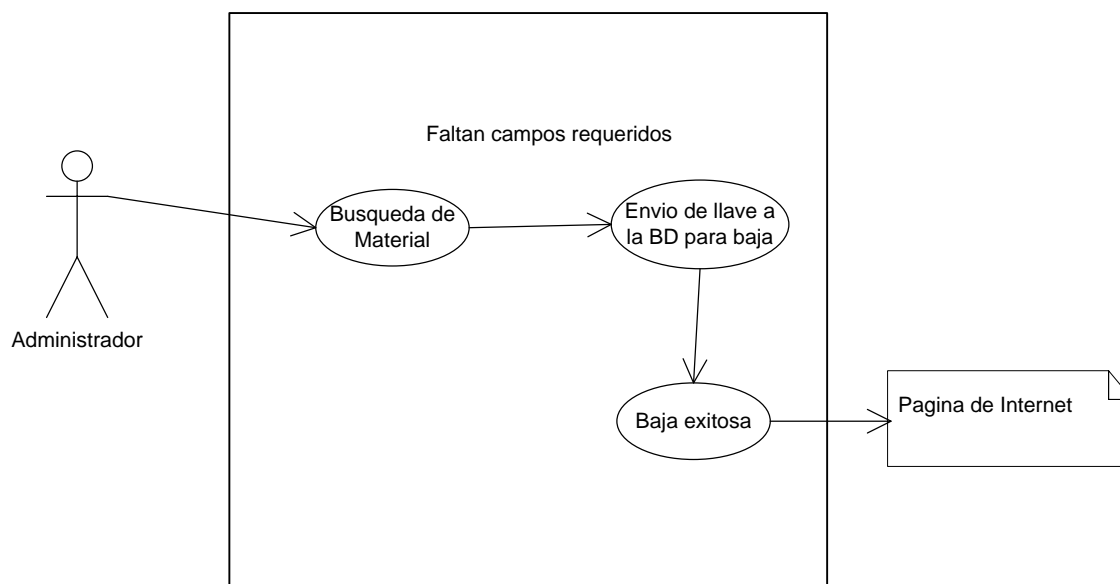
CASO DE USO: OBTENER REPORTE ESTADISTICOS	
ACTOR: ADMINISTRADOR	
CURSO NORMAL	ALTERNATIVA
1) El administrador elegirá la opción Reportes del menú principal.	
2) El sistema obtendrá la información de la Base de Datos y calculara el reporte correspondiente.	2.1) La información del reporte estará limitada a la cantidad de operaciones realizadas hasta el momento en el sistema.
3) El administrador vera entonces el reporte listo para imprimir en la pantalla	

Diagrama de casos de uso:

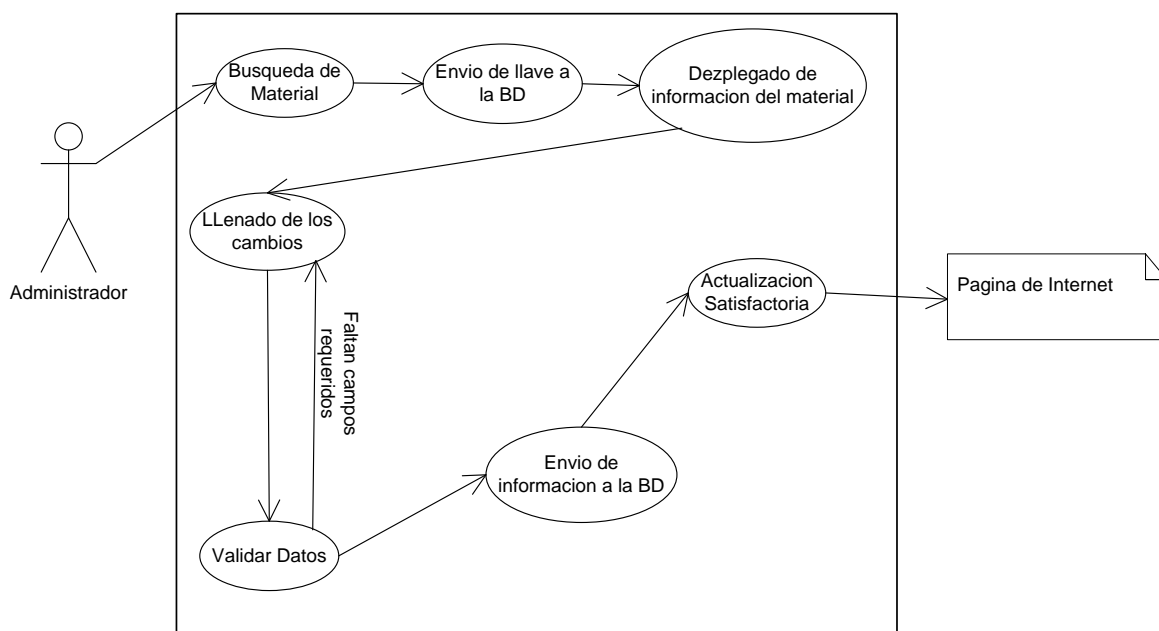




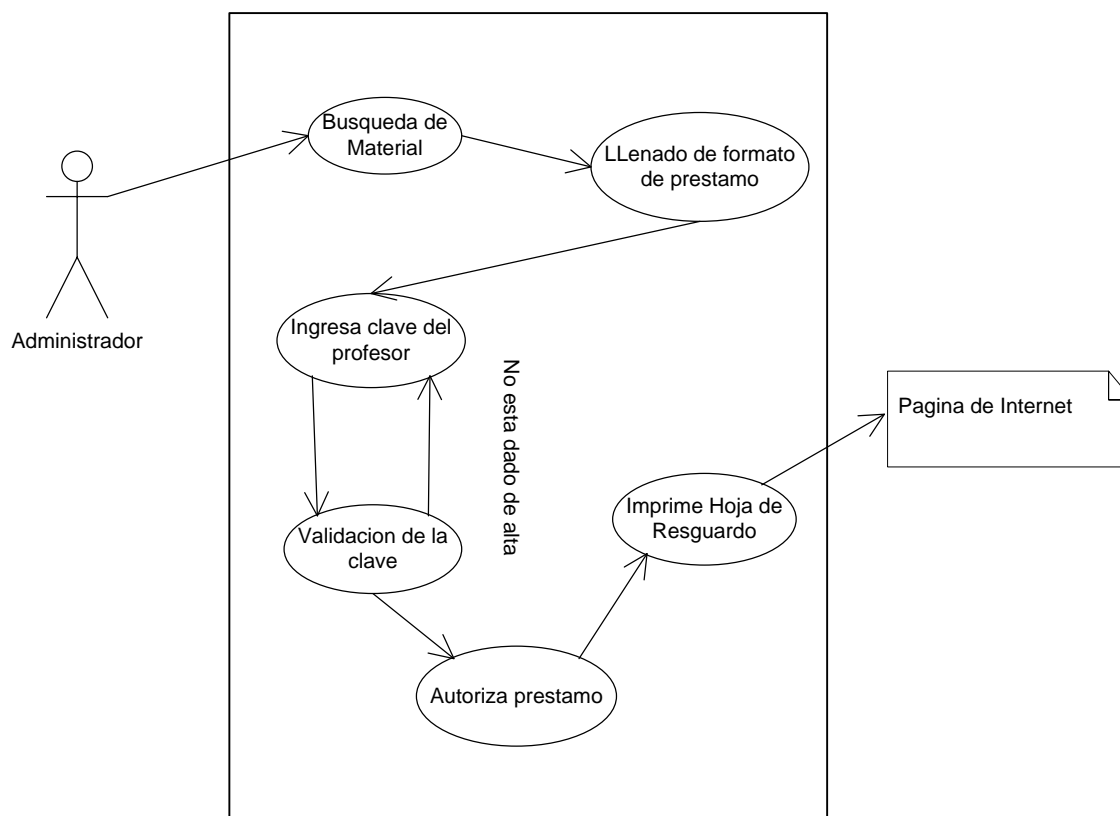
Nivel 3.1



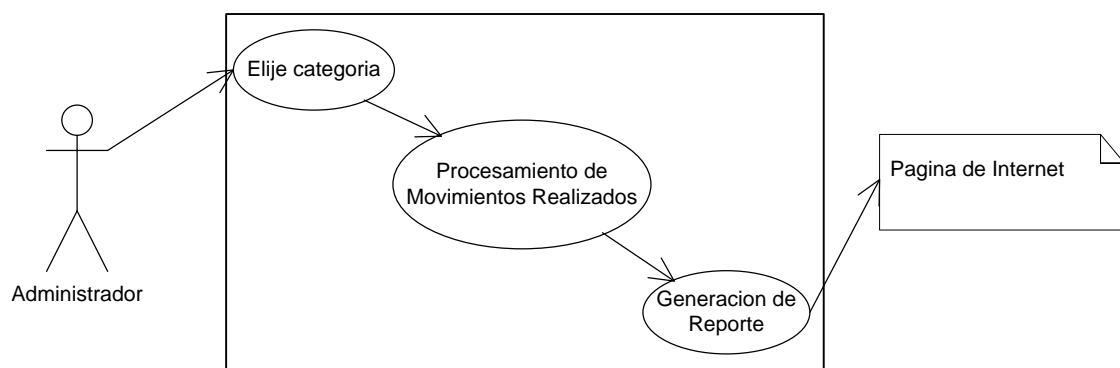
Nivel 4.1

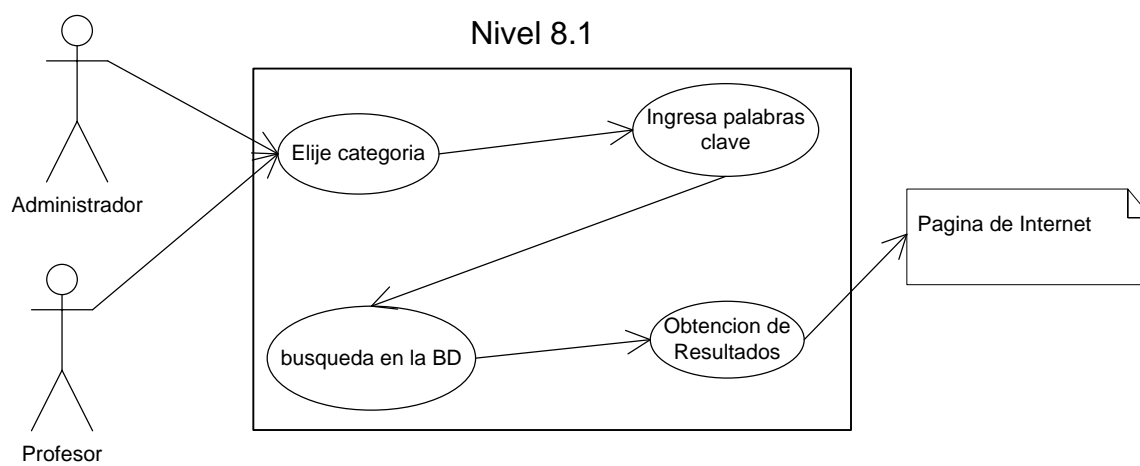
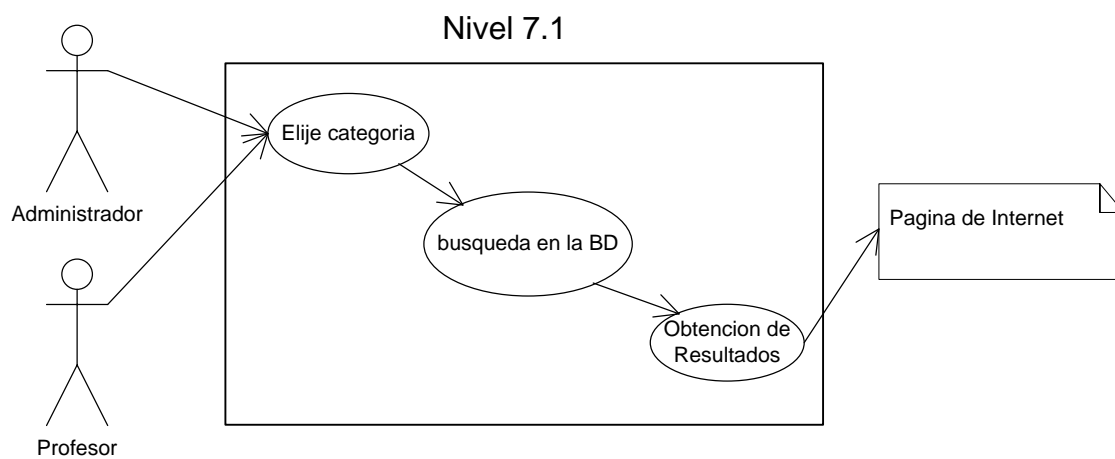


Nivel 5.1



Nivel 6.1

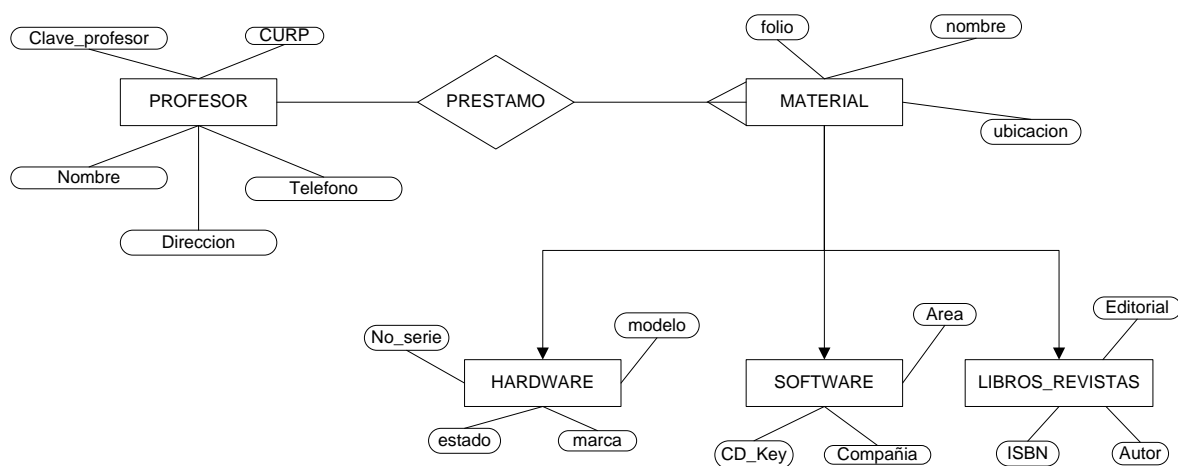




2.3 DISEÑO

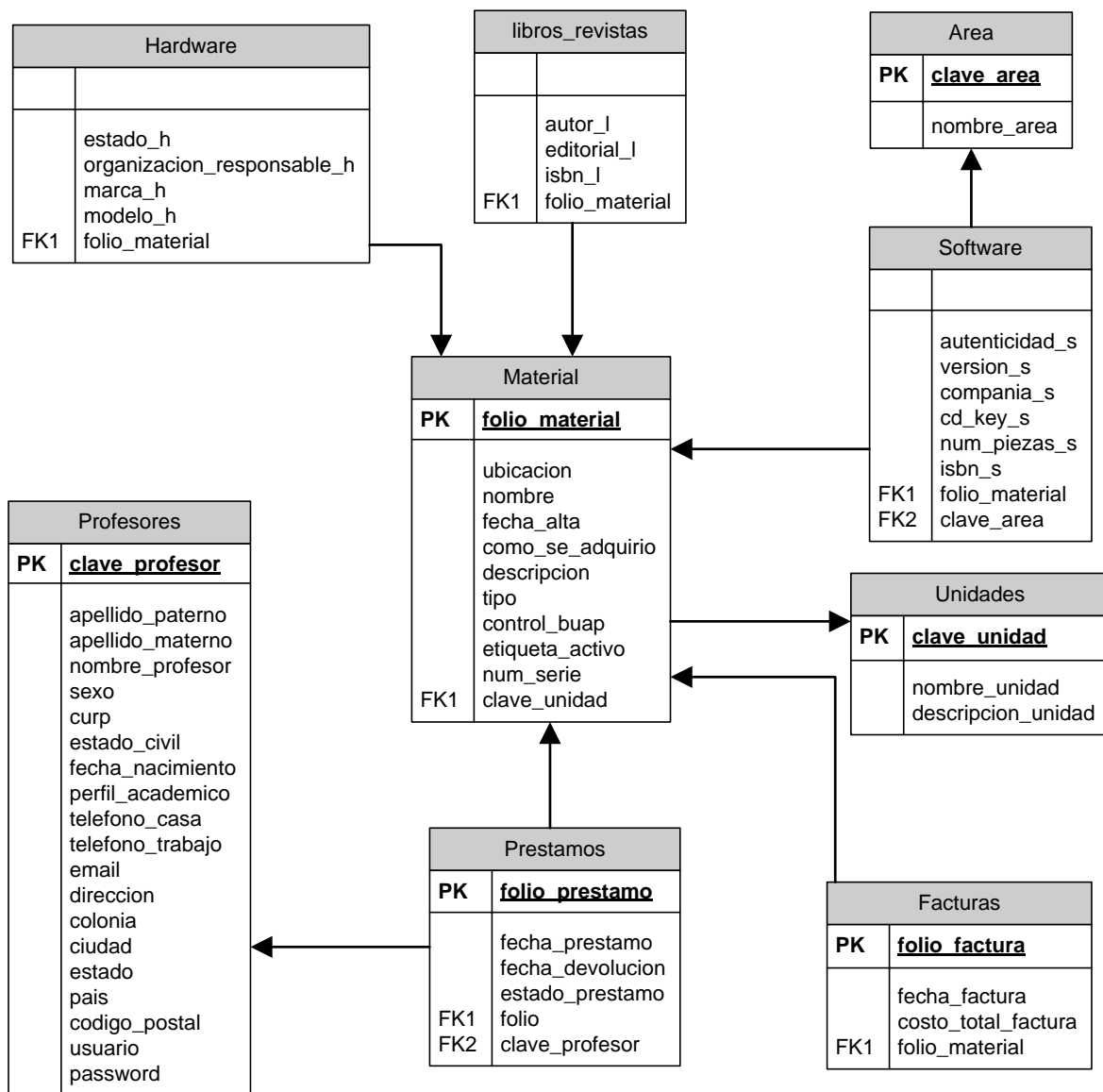
Dentro de este apartado, se abordará principalmente el tema de la dependencia funcional de los datos con la finalidad de obtener un esquema de Bases de datos con buenas propiedades y manteniendo la integridad de la misma. Para ello, el presente diseño está basado en la teoría de las dependencias funcionales y de las formas normales para las relaciones.

2.3.1 MODELO ENTIDAD -RELACION



2.3.2 MODELO RELACIONAL DE LA BASE DE DATOS

A continuación se muestran las tablas, relaciones y atributos resultantes después de aplicar las técnicas de identificación de dependencias funcionales y normalización.



2.3.3 DICCIONARIO DE DATOS: TABLAS Y CAMPOS

Tabla: Profesores

Descripción: Permite la administración de la información personal de profesores y cuentas de usuario.

CAMPOS	
NOMBRE	DESCRIPCION
clave_profesor	Llave primaria para la tabla
apellido_paterno	Apellido paterno del usuario
apellido_materno	Apellido materno del usuario
nombre_profesor	Nombre(s) del profesor
sexo	Sexo H=hombre, M=mujer
curp	Clave Única de Registro De Población
estado_civil	S=Soltero, C=casado, UL=unión libre
fecha_nacimiento	Fecha de nacimiento del profesor
perfil_academico	Nivel de preparación
teléfono_casa	Teléfono de domicilio
teléfono_trabajo	Teléfono de oficina de trabajo
email	Dirección de correo electrónico personal
dirección	Dirección de domicilio del profesor
colonia	Colonia de domicilio del profesor
ciudad	Ciudad de domicilio del profesor
estado	Estado al que pertenece
pais	Pais al que pertenece
código_postal	Código postal del domicilio del profesor
usuario	Nombre de usuario del acceso al sistema
password	Contraseña de acceso al sistema

Tabla: Material

Descripción: Permite la administración de la información del material existente en inventario.

CAMPOS	
NOMBRE	DESCRIPCION
ubicación	Descripción del lugar donde se encuentra el material
nombre	Nombre del material
fecha_alta	Fecha de alta del material al inventario
como_se_adquirio	1=compra,2=donación
descripción	Descripción breve del material
tipo	1=hardware,2=software,3=libro o revista
control_buap	Numero de control de la universidad
etiqueta_activo	Etiqueta de control de la universidad
num_serie	Numero de serie del material
clave_unidad	Numero <u>de</u> referencia a la tabla unidades

Tabla: Hardware

Descripción: Permite la administración de la información del material de tipo hardware existente en inventario.

CAMPOS	
NOMBRE	DESCRIPCION
estado_h	Descripción breve del estado del hardware
organización_responsable	Nombre de la organización a cargo del hardware
marca_h	Marca del Hardware
modelo_h	Modelo del Hardware
folio_material	Llave foránea de acceso al material de este tipo

Tabla: libros_revistas

Descripción: Permite la administración de la información del material de tipo libros/revistas existente en inventario.

CAMPOS	
NOMBRE	DESCRIPCION
autor_l	Nombre del autor del libro/revista
editorial_l	Nombre de la editorial que emite el ejemplar.
isbn_l	ISBN del libro/revista si existe.
folio_material	Llave foránea de acceso al material de este tipo

Tabla: Software

Descripción: Permite la administración de la información del material de tipo software existente en inventario.

CAMPOS	
NOMBRE	DESCRIPCION
autenticidad_s	Indica si es original o copia
versión_s	Versión del Software
compañía_s	Compañía que desarrollo el software
cd_key_s	Numero llave de instalación del software
num_piezas_s	Numero de discos en que viene el software
isbn_s	Numero ISBN si es que lo tiene.
folio_material	Llave foránea de acceso al material de este tipo
clave_area	Llave que nos indica en la tabla áreas la categoría a la que pertenece el software.

Tabla: Area

Descripción: Permite la administración de la información áreas o categorías para el software.

CAMPOS	
NOMBRE	DESCRIPCION
Clave_area	Llave primaria de la tabla
Nombre_area	Nombre del área o categoría para el software

Tabla: Unidades

Descripción: Permite la administración de la información unidades en las que puede presentarse la adquisición de material.

CAMPOS	
NOMBRE	DESCRIPCION
clave_unidad	Llave primaria de la tabla
nombre_unidad	Nombre de la unidad. Por ejemplo: pieza, paquete, etc.
descripción_unidad	Descripción de la unidad en caso de ser de ser necesaria

Tabla: Facturas

Descripción: Permite la administración de facturas de compra de material.

CAMPOS	
NOMBRE	DESCRIPCION
folio_factura	Folio correspondiente a la factura y al mismo tiempo llave de la tabla
costo_total_factura	Indica el costo total de compra de la factura
folio_material	Llave foránea que indica a que material corresponde la factura.

Tabla: Prestamos

Descripción: Permite la administración de préstamos de material a profesores.

CAMPOS	
NOMBRE	DESCRIPCION
folio_prestamo	Llave de la tabla de prestamos
fecha_prestamo	Fecha en que se presta el material
fecha_devolucion	Fecha en que se devuelve el material
estado_prestamo	Si aun sigue en préstamo (1) o si está disponible (0).
folio	Llave foránea que Indica el material en préstamo
clave_profesor	Llave foránea que Indica el profesor que solicito el material.

CAPITULO 3 – IMPLEMENTACION

3.1 INTRODUCCION

Dentro de este capítulo, se abordará principalmente la implementación del sistema. Las herramientas utilizadas para el desarrollo del sistema fueron:

- Microsoft Access
- Bloc de Notas
- ASP
- HTML
- Sistema Operativo Windows Server 2003
- IIS

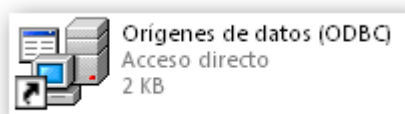
3.2 CONECTIVIDAD A LA BASE DE DATOS

ASP proporciona un método sencillo pero eficiente para crear sitios web dinámicos y con acceso a bases de datos.

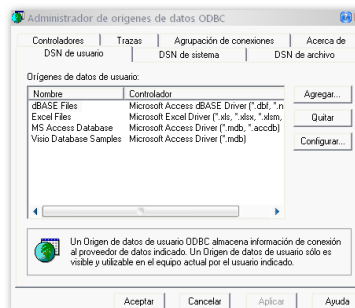
Para llevar a cabo la conectividad es necesario conocer la ruta de la base de datos, y un método fácil para ubicarla es contar con un DNS (Domain Name System). Hay que resaltar que el DNS no interactúa con la base de datos, es decir, solo nos dirá donde se encuentra, pero podemos hacerlo mediante objetos ADODB una vez que conozcamos su ubicación.

Para crear un DNS el procedimiento es el siguiente:

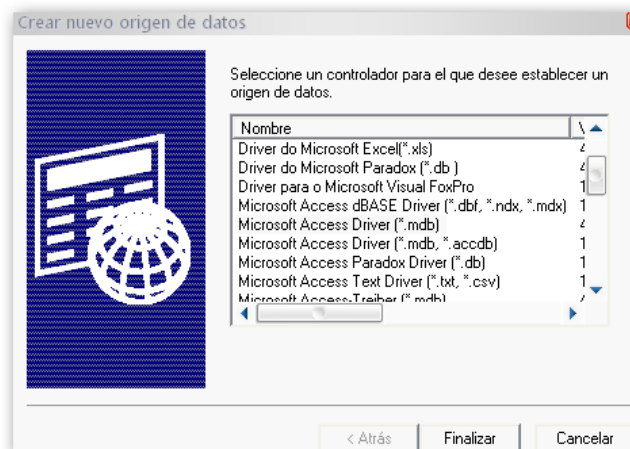
- Ir al *Menú inicio -> Configuración -> Panel de control -> Orígenes de Datos ODBC*.



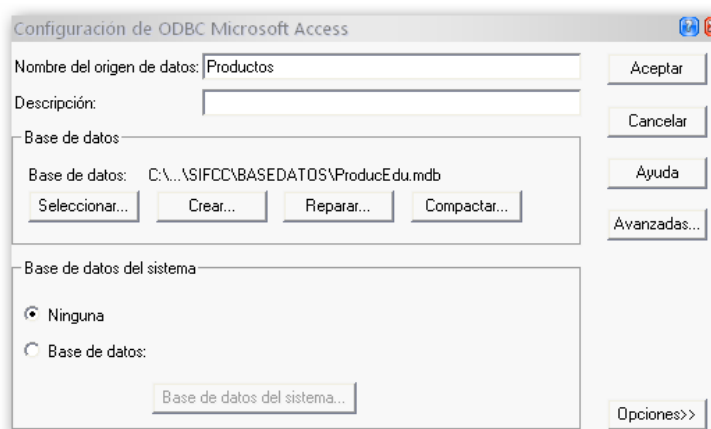
- Y se desplegará la siguiente ventana:



- Una vez estando en este paso, hay que crear un DNS de sistema, para ello vamos a la pestaña correspondiente y elegimos el botón Agregar y buscamos el controlador de Microsoft Access Driver.



- Finalmente, hay que establecer el nombre que utilizaremos en programación para acceder a la base de datos y elegir también la ruta donde se encuentra ubicada la base de datos en el disco local.



3.3 DESARROLLO DEL SISTEMA

La implementación del sistema fue desarrollada primeramente pensando en el administrador del sistema. El sitio web está diseñado y programado para autenticar al administrador en una base de datos creada en Access y le permite realizar operaciones sobre la misma mediante la interacción con páginas escritas en lenguaje ASP (similar al lenguaje Visual Basic). Contiene principalmente los siguientes módulos:

- Hardware (Altas, Bajas, Modificaciones, Prestamos).
- Software (Altas, Bajas, Modificaciones, Prestamos).
- Libros y Revistas (Altas, Bajas, Modificaciones, Prestamos).
- Búsquedas de material
- Reportes.

3.4 CODIGO FUENTE

Solo para ejemplificar, muestro la codificación empleada para las operaciones pertinentes para el hardware existente en el inventario:

Acciones hardware.asp

```
<HTML>
<HEAD>
<TITLE>Acciones para el hardware</TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="scripts/FSdateSelect.js"></SCRIPT>
  <LINK REL="stylesheet" HREF="styles/FSdateSelect.css" type="text/css">
</HEAD>
<%@ LANGUAGE=JScript %>
<BODY>
<p>
  <!-- #INCLUDE File="ADOJAVAS.inc" -->
  <Script language="JavaScript">
<!--

function ApareceAyuda(){
  ayuda.style.visibility='visible';
  return true;
}

function confirmar_reset(){
  return confirm("¿ Deseas Limpiar el formulario... ?");
}

function confirmar_submit(){
  return confirm("¿ Deseas enviar el formulario... ?");
}

function aMayusculas(campo){
  campo.value=campo.value.toUpperCase();
}

function cambiaEstado(texto){
  window.status=texto;
}

function compruebaFecha(campo){
  var array_pal= new Array();
  array_pal=campo.value.split("/");
  if (array_pal.length ==3)
  {
    if (parseInt(array_pal[1])>12)
    {
      alert("El mes debe ser menor que 12\nEl formato es Día/Mes/Año");
      campo.focus();
      campo.select();
    }
    if (parseInt(array_pal[0])>32)
    {
      alert("El día debe ser menor o igual que 31\nEl formato es
Día/Mes/Año");
      campo.focus();
      campo.select();
    }
  }
} else
{
  alert("El formato es Día/Mes/Año\n Ejemplo 5/12/2002");
  campo.focus();
  campo.select();
}
}
```

```

    }
//-->
</Script>

<%
Fondo=parseInt(Request.QueryString ("Fondo"))
Boton=Request.QueryString ("Boton")
Response.Cookies ("folio")=Fondo

if (Boton=="Modificar")
{
    Ob_Conn = new ActiveXObject ("ADODB.Connection")
    Ob_Command = new ActiveXObject ("ADODB.Command")
    Ob_Conn.Open ("Productos")
    Ob_Command.ActiveConnection = Ob_Conn
    Ob_Command.CommandType = adCmdText
    Sql= "SELECT * FROM hardware WHERE folio= "+Fondo
    Ob_Command.CommandText = Sql
    Ob_RS = Ob_Command.Execute ()
    if (Ob_RS.EOF)
    {
        Response.Write("No hay Estudios Realizados")
    }
}
%>
<FORM METHOD="Post" Name="miForm" ACTION="ModifyHardware.asp">
</p>

<table width="52%" border="1" align="center" cellpadding="1" cellspacing="1"
bordercolor="#CC9900" bgcolor="#990033">
<tr>
<td height="24"><div align="center"><strong><font color="#CC9900"
size="4">Modificar Datos de Hardware</font></strong></div></td>
</tr>
</table>

<table width="75%" border="0" align="center">
<tr>
<td width="42%"><INPUT Name="text_folio" Type="Text" id="text_folio"
style="visibility:hidden" value="<%=Ob_RS("folio")%>" Size=10 MaxLength=10
"></td>
<td width="45%">&nbsp;</td>
</tr>
<tr>
<td><B>Nombre:</B></td>
<td width="45%"><INPUT Name="text_nombre" Type="Text" id="text_nombre"
onChange="aMayusculas(this)" value="<%=Ob_RS("nombre")%>" Size=70 MaxLength=70
"></td>
</tr>
<tr>
<td><strong>Descripci&oacute;n:</strong></td>
<td width="45%"><INPUT Name="text_descripcion" Type="Text"
id="text_descripcion" onChange="aMayusculas(this)"
value="<%=Ob_RS("descripcion")%>" Size=70 MaxLength=70 ></td>
<td width="13%">&nbsp;</td>
</tr>
<tr>
<td><strong>Control buap:</strong></td>
<td width="45%"><INPUT Name="text_controlbuap" Type="Text"
id="text_controlbuap" onChange="aMayusculas(this)"
value="<%=Ob_RS("controlbuap")%>" Size=70 MaxLength=70 ></td>
<td>&nbsp;</td>
</tr>

```

```

<tr>
  <td><strong>Ubicaci&oacute;n:</strong></td>
  <td><INPUT Name="text_ubicacion" Type="Text" id="text_ubicacion"
onChange="aMayusculas(this)" value="<%=Ob_RS("ubicacion")%" Size=70 MaxLength=70
></td>
</tr>
<%
  var_estado1= 1
  cadena_estado1="BUENAS CONDICIONES"
  var_estado2= 2
  cadena_estado2="DESCOMPUESTO"
  var_estado3= 3
  cadena_estado3="REPARACI&Oacute;N"
  var_estado4= 4
  cadena_estado4="BAJA"
  if (Ob_RS("estado") == 1)
  {
    var_estado1= 1
    cadena_estado1="BUENAS CONDICIONES"
    var_estado2= 2
    cadena_estado2="DESCOMPUESTO"

  }

  if (Ob_RS("estado") == 2) {
    var_estado1= 2
    cadena_estado1="DESCOMPUESTO"
    var_estado2= 1
    cadena_estado2="BUENAS CONDICIONES"
  }

  if (Ob_RS("estado") == 3) {
    var_estado1= 3
    cadena_estado1="REPARACI&Oacute;N"
    var_estado3= 1
    cadena_estado3="BUENAS CONDICIONES"
  }

  if (Ob_RS("estado") == 4) {
    var_estado1= 4
    cadena_estado1="BAJA"
    var_estado4= 1
    cadena_estado4="BUENAS CONDICIONES"
  }
%>

<tr>
  <td><b>Estado:</b></td>
  <td><select name="text_estado">
    <option value=<%=var_estado1%> selected> <%=cadena_estado1%> </option>
    <option value=<%=var_estado2%>><%=cadena_estado2%> </option>
    <option value=<%=var_estado3%>><%=cadena_estado3%> </option>
    <option value=<%=var_estado4%>><%=cadena_estado4%> </option>
  </select></td>
</tr>

<%
  v1=""<%=Ob_RS("dia_alta");
  v2=""<%=Ob_RS("mes_alta");
  v3=""<%=Ob_RS("ano_alta");
%>

<tr>

```

```

<td><strong>Fecha de Alta: </strong></td>
<td><B>Día:</B>

    <SELECT Name="text_dia_alta">
    <%if (v1=="null")
    {
    %>
    <OPTION Value=0 SELECTED> Día </OPTION>
    <%
    }
    else
    {
    %>
    <OPTION Value=<%=Ob_RS("dia_alta")%> SELECTED>
<%=Ob_RS("dia_alta")%></OPTION>
    <%}
    for (i=1;i<=31;i++)
    { %>
    <OPTION Value=<%=i%>> <%=i%></OPTION>
    <% } %>
    </SELECT>

    <B>Mes</B>
    <%if (v2=="null")
    {%>

    <select name="text_mes_alta">
    <option value=0 selected> Mes </option>
    <%}
    else
    {%>
    <select name="text_mes_alta" >
    <option value= <%=Ob_RS("mes_alta")%> selected> <%=Ob_RS("mes_alta")%>
</option>
    <%}
    for (i=1;i<=12;i++)
    { %>
    <option value=<%=i%>> <%=i%></option>
    <% } %>
    </select>

    <B> Año</B>
    <%if (v3=="null")
    {%>
    <SELECT Name="text_ano_alta">
    <OPTION Value=0 SELECTED> Año </OPTION>
    <%}
    else
    {%>
    <SELECT Name="text_ano_alta">
    <OPTION Value= <%=Ob_RS("ano_alta")%> SELECTED> <%=Ob_RS("ano_alta")%>
</OPTION>
    <%}
    for (i=1965;i<=2007;i++)
    { %>
    <OPTION Value=<%=i%>> <%=i%></OPTION>
    <% } %>
    </SELECT>    </td>

    <td>&nbsp;</td>
</tr>

<tr>

```

```

        <td><strong>Etiqueta Activo:</strong></td>
        <td><input name="text_etiqueta_activo" type="Text"
onChange="aMayusculas(this)" value="<%=Ob_RS("etiqueta_activo")%>" size=70
maxlength=70 ></td>
    <td>&nbsp;</td>
</tr>

<%
    v1="" +Ob_RS("dia_compra");
    v2="" +Ob_RS("mes_compra");
    v3="" +Ob_RS("ano_compra");
%>

<tr>
    <td><strong>Fecha de Compra: </strong></td>
    <td><B>Día:</B>
        <SELECT Name="text_dia_compra">
            <%if (v1=="null")
            {
            %>
                <OPTION Value=0 SELECTED> Día </OPTION>
            <%
            }
            else
            {
            %>
                <OPTION Value=<%=Ob_RS("dia_compra")%> SELECTED>
<%=Ob_RS("dia_compra")%></OPTION>
            <%
            for (i=1;i<=31;i++)
            { %>
                <OPTION Value=<%=i%>> <%=i%></OPTION>
            <% } %>
            </SELECT>

            <B>Mes</B>
            <%if (v2=="null")
            {%>

                <select name="text_mes_compra">
                    <option value=0 selected> Mes </option>
                <%}
            else
            {%>
                <select name="text_mes_compra" >
                    <option value= <%=Ob_RS("mes_compra")%> selected>
<%=Ob_RS("mes_compra")%> </option>
                <%}
                for (i=1;i<=12;i++)
                { %>
                    <option value=<%=i%>> <%=i%></option>
                <% } %>
            </select>

            <B> Año</B>
            <%if (v3=="null")
            {%>
                <SELECT Name="text_ano_compra">
                    <OPTION Value=0 SELECTED> Año </OPTION>
                <%}
            else
            {%>
                <SELECT Name="text_ano_compra">
                    <OPTION Value= <%=Ob_RS("ano_compra")%> SELECTED>
<%=Ob_RS("ano_compra")%> </OPTION>

```

```

        <%}
        for (i=1965;i<=2007;i++)
        { %>
            <OPTION Value=<%=i%>> <%=i%></OPTION>
            <% } %>
        </SELECT>        </td>

        <td>&nbsp;</td>
    </tr>

    <tr>
        <td><strong>Costo Total:</strong></td>
        <td><input name="text_costo_total" type="Text" onChange="aMayusculas(this)"
value="<%=Ob_RS("costo_total")%>" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>N&uacute;mero de Factura:</strong></td>
        <td><input name="text_no_factura" type="Text" id="text_no_factura"
onChange="aMayusculas(this)" value="<%=Ob_RS("no_factura")%>" size=70
maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Organizaci&oacute;n Responsable :</strong></td>
        <td><input name="text_organizacion_responsable" type="Text"
onChange="aMayusculas(this)" value="<%=Ob_RS("organizacion_responsable")%>"
size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Unidad:</strong></td>
        <td><input name="text_unidad" type="Text" id="text_unidad"
onChange="aMayusculas(this)" value="<%=Ob_RS("unidad")%>" size=70 maxlength=70
></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Endosar a :</strong></td>
        <td><input name="text_endosar_a" type="Text" id="text_endosar_a"
onChange="aMayusculas(this)" value="<%=Ob_RS("endosar_a")%>" size=70 maxlength=70
></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Como se adquirio:</strong></td>
        <td><select name="select">
            <option value="1">Compra</option>
            <option value="2">Donacion</option>
        </select>
        </td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Marca :</strong></td>
        <td><input name="text_marca" type="Text" id="text_marca"
onChange="aMayusculas(this)" value="<%=Ob_RS("marca")%>" size=70 maxlength=70
></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Modelo:</strong></td>

```

```

        <td><input name="text_modelo" type="Text" id="text_modelo"
onChange="aMayusculas(this)" value="<%=Ob_RS("modelo")%">" size=70 maxlength=70
></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>No. serie :</strong></td>
        <td><input name="text_no_serie" type="Text" id="text_no_serie"
onChange="aMayusculas(this)" value="<%=Ob_RS("no_serie")%">" size=70 maxlength=70
></td>
        <td>&nbsp;</td>
    </tr>
</table>
<table width="75%" border="0" align="center">
    <tr>
        <td width="3%"><input name="Submit" type="Submit" value="Enviar" > </td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>
<p>&nbsp;</p>
</FORM>

<%}%>

<%
if (Boton=="Nuevo")
{

%>
<FORM METHOD="Post" Name="miform" ACTION="nuevo_hardware.asp">
</p>

<table width="52%" border="1" align="center" cellpadding="1" cellspacing="1"
bordercolor="#CC9900" bgcolor="#990033">
    <tr>
        <td height="24"><div align="center"><strong><font color="#CC9900"
size="4">Ingresar Datos de Nuevo Hardware</font></strong></div></td>
    </tr>
</table>

<table width="75%" border="0" align="center">
    <tr>
        <td><B>Nombre:</B></td>
        <td width="45%"><INPUT Name="text_nombre" Type="Text" id="text_nombre"
onChange="aMayusculas(this)" Size=70 MaxLength=70 ></td>
    </tr>
    <tr>
        <td><strong>Descripci&oacute;n:</strong></td>
        <td width="45%"><INPUT Name="text_descripcion" Type="Text"
id="text_descripcion" onChange="aMayusculas(this)" Size=70 MaxLength=70 ></td>
        <td width="13%">&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Control buap:</strong></td>
        <td width="45%"><INPUT Name="text_controlbuap" Type="Text"
id="text_controlbuap" onChange="aMayusculas(this)" Size=70 MaxLength=70 ></td>
        <td>&nbsp;</td>
    </tr>

```

```

<tr>
  <td><strong>Ubicaci&oacute;n:</strong></td>
  <td><INPUT Name="text_ubicacion" Type="Text" id="text_ubicacion"
onChange="aMayusculas(this)" Size=70 MaxLength=70 ></td>
  <td>&nbsp;</td>
</tr>

<%
  var_estado1= 1
  cadena_estado1="BUENAS CONDICIONES"
  var_estado2= 2
  cadena_estado2="DESCOMPUESTO"
  var_estado3= 3
  cadena_estado3="REPARACI&Oacute;N"
  var_estado4= 4
  cadena_estado4="BAJA"

%>

<tr>
  <td><b>Estado:</b></td>
  <td><select name="text_estado">
    <option value=<%=var_estado1%> selected> <%=cadena_estado1%> </option>
    <option value=<%=var_estado2%>><%=cadena_estado2%> </option>
    <option value=<%=var_estado3%>><%=cadena_estado3%> </option>
    <option value=<%=var_estado4%>><%=cadena_estado4%> </option>
  </select></td>
</tr>

<tr>
  <td><strong>Fecha de Alta: </strong></td>
  <td><B>D&iacute;a:</B>
    <SELECT Name="text_dia_alta">
      <OPTION Value=0 SELECTED> D&iacute;a </OPTION>
    <%
      for (i=1;i<=31;i++)
      { %>
        <OPTION Value=<%=i%>> <%=i%></OPTION>
      <% } %>
    </SELECT>

    <B>Mes</B>
    <select name="text_mes_alta">
      <option value=0 selected> Mes </option>
    <%
      for (i=1;i<=12;i++)
      { %>
        <option value=<%=i%>> <%=i%></option>
      <% } %>
    </select>

    <B> A&ntilde;o</B>
    <SELECT Name="text_ano_alta">
      <OPTION Value=0 SELECTED> A&ntilde;o </OPTION>
    <%
      for (i=1965;i<=2007;i++)
      { %>
        <OPTION Value=<%=i%>> <%=i%></OPTION>
      <% } %>
    </SELECT>      </td>

  <td>&nbsp;</td>
</tr>
<tr>
  <td><strong>Etiqueta Activo:</strong></td>

```

```

        <td><input name="text_etiqueta_activo" type="Text"
onChange="aMayusculas(this)" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Fecha de Compra: </strong></td>
        <td><B>Día:</B>
            <SELECT Name="text_dia_compra">
                <OPTION Value=0 SELECTED> Día </OPTION>
                <%
                    for (i=1;i<=31;i++)
                    { %>
                        <OPTION Value=<%=i%>> <%=i%></OPTION>
                    <% } %>
            </SELECT>
            <B>Mes</B>
            <select name="text_mes_compra">
                <option value=0 selected> Mes </option>
                <%
                    for (i=1;i<=12;i++)
                    { %>
                        <option value=<%=i%>> <%=i%></option>
                    <% } %>
            </select>

            <B> Año</B>
            <SELECT Name="text_ano_compra">
                <OPTION Value=0 SELECTED> Año </OPTION>
                <%
                    for (i=1965;i<=2007;i++)
                    { %>
                        <OPTION Value=<%=i%>> <%=i%></OPTION>
                    <% } %>
            </SELECT>          </td>

        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Costo Total:</strong></td>
        <td><input name="text_costo_total" type="Text" onChange="aMayusculas(this)"
size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>N&uacute;mero de Factura:</strong></td>
        <td><input name="text_no_factura" type="Text" id="text_no_factura"
onChange="aMayusculas(this)" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Organizaci&oacute;n Responsable :</strong></td>
        <td><input name="text_organizacion_responsable" type="Text"
onChange="aMayusculas(this)" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Unidad:</strong></td>
        <td><input name="text_unidad" type="Text" id="text_unidad"
onChange="aMayusculas(this)" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Endosar a :</strong></td>

```

```

        <td><input name="text_endosar_a" type="Text" id="text_endosar_a"
onChange="aMayusculas(this)" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Como se adquirio :</strong></td>
        <td><select name="text_como_se_adquirio" id="text_como_se_adquirio">
            <option value="1">Compra</option>
            <option value="2">Donacion</option>
        </select> </td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Marca :</strong></td>
        <td><input name="text_marca" type="Text" id="text_marca"
onChange="aMayusculas(this)" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Modelo:</strong></td>
        <td><input name="text_modelo" type="Text" id="text_modelo"
onChange="aMayusculas(this)" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>No. serie :</strong></td>
        <td><input name="text_no_serie" type="Text" id="text_no_serie"
onChange="aMayusculas(this)" size=70 maxlength=70 ></td>
        <td>&nbsp;</td>
    </tr>
</table>
<table width="75%" border="0" align="center">
    <tr>
        <td width="3%"><input name="Submit" type="Submit" value="Enviar" > </td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>
<p>&nbsp;</p>
</FORM>

<%>

if (Boton=="Borrar")
{
    sql="DELETE FROM hardware WHERE folio="+ Fondo
    Ob_Conn = new ActiveXObject ("ADODB.Connection")
    Ob_Command = new ActiveXObject ("ADODB.Command")
    Ob_Conn.Open ("Productos")
    Ob_Command.ActiveConnection = Ob_Conn
    Ob_Command.CommandType = adCmdText
    Ob_Command.CommandText = sql
    Ob_Command.Execute()
    Ob_Conn.Close()
    %> <script>alert('Eliminacion Exitosa');</script> <%
    Response.Redirect ("PrincipalHardware.asp")
}

%>

```

```

<%
if (Boton=="Prestamo")
{
    Ob_Conn = new ActiveXObject ("ADODB.Connection")
    Ob_Command = new ActiveXObject ("ADODB.Command")
    Ob_Conn.Open ("Productos")
    Ob_Command.ActiveConnection = Ob_Conn
    Ob_Command.CommandType = adCmdText
    Sql= "SELECT * FROM hardware WHERE folio= "+Fondo
    Ob_Command.CommandText = Sql
    Ob_RS = Ob_Command.Execute ()

    %>
<table width="52%" border="1" align="center" cellpadding="1" cellspacing="1"
bordercolor="#CC9900" bgcolor="#990033">
    <tr>
        <td height="24"><div align="center"><strong><font color="#CC9900"
size="4">Hoja de Prestamo</font></strong></div></td>
    </tr>
</table>
<form action="prestamo_hardware.asp" method="post" name="nuevoPrestamo"
target="_self" id="nuevoPrestamo">
<p>&nbsp;</p>
<table width="68%" border="0" align="center">
    <tr><tr>
        <td><B>Clave de Control:</B></td>
        <td><input name="text_folio" type="Text" id="text_folio"
onChange="aMayusculas(this)" value="<%=Ob_RS("folio")%>" size=70 maxlength=70
></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td width="45%"> </tr>
    <tr>
        <td><B>Nombre:</B></td>
        <td width="45%"><%=Ob_RS("nombre")%> </tr>
    <tr>
        <td><strong>Descripci&oacute;n:</strong></td>
        <td width="45%"><%=Ob_RS("descripcion")%>
        <td width="13%">&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Control buap:</strong></td>
        <td width="45%"><%=Ob_RS("controlbuap")%>
        <td>&nbsp;</td>
    </tr>

    <tr>
        <td><strong>Marca :</strong></td>
        <td><%=Ob_RS("marca")%></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><strong>Modelo:</strong></td>
        <td><%=Ob_RS("modelo")%></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td height="21"><strong>No. serie :</strong></td>
        <td><%=Ob_RS("no_serie")%></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td height="25"><strong>Fecha Prestamo:</strong></td>
        <td height="25"><B>D&iacute;a:</B>
            <SELECT Name="text_dia_prestamo" id="text_dia_prestamo">

```

```

        <OPTION Value=0 SELECTED> D&iacute;a </OPTION>
        <%
        for (i=1;i<=31;i++)
        { %>
        <OPTION Value=<%=i%>> <%=i%></OPTION>
        <% } %>
</SELECT>
<B>Mes</B>
<select name="text_mes_prestamo" id="select2">
  <option value=0 selected> Mes </option>
  <%
  for (i=1;i<=12;i++)
  { %>
  <option value=<%=i%>> <%=i%></option>
  <% } %>
</select>
<B> A&ntilde;o</B>
<SELECT Name="text_ano_prestamo" id="select3">
  <OPTION Value=0 SELECTED> A&ntilde;o </OPTION>
  <%
  for (i=1965;i<=2007;i++)
  { %>
  <OPTION Value=<%=i%>> <%=i%></OPTION>
  <% } %>
</SELECT></td>
<td height="25"><strong></strong></td>
</tr>
<tr>
<td height="25"><strong>Fecha Entrega: </strong></td>
<td><B>D&iacute;a:</B>
  <SELECT Name="text_dia_devolucion" id="text_dia_devolucion">
    <OPTION Value=0 SELECTED> D&iacute;a </OPTION>
    <%
    for (i=1;i<=31;i++)
    { %>
    <OPTION Value=<%=i%>> <%=i%></OPTION>
    <% } %>
  </SELECT>
  <B>Mes</B>
  <select name="text_mes_devolucion" id="text_mes_devolucion">
    <option value=0 selected> Mes </option>
    <%
    for (i=1;i<=12;i++)
    { %>
    <option value=<%=i%>> <%=i%></option>
    <% } %>
  </select>
  <B> A&ntilde;o</B>
  <SELECT Name="text_ano_devolucion" id="text_ano_devolucion">
    <OPTION Value=0 SELECTED> A&ntilde;o </OPTION>
    <%
    for (i=1965;i<=2007;i++)
    { %>
    <OPTION Value=<%=i%>> <%=i%></OPTION>
    <% } %>
  </SELECT> </td>
<td height="25"><strong></strong></td>
</tr>
<tr>
  <td><strong>Nombre</strong></td>
  <td><select name="text_clave_profesor" id="text_clave_profesor">
  <%
  Sql2= "SELECT clave_profesor,nombre,apellido_paterno,apellido_materno FROM
datos_personales order by apellido_paterno"

```

```

Ob_Command.CommandText = Sql2
Ob_RS2 = Ob_Command.Execute ()

while (!Ob_RS2.Eof){
%>
    <option value="<%= Ob_RS2("clave_profesor") %>"> <%=
Ob_RS2("apellido_paterno") %>&nbsp;<%= Ob_RS2("apellido_materno") %>&nbsp;<%=
Ob_RS2("nombre") %> </option>
    <%
    Ob_RS2.MoveNext()
    }
    Ob_Conn.Close()
    %>
    </select>    </td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td><strong>Firma</strong></td>
    <td>_____</td>
</tr>
</table>
<p align="center">
    <input name="Confirmar" type="submit" id="Confirmar" value="Confirmar">
</p>
</form>
<p>
<div align="justify"><div align="left">
    <p align="center"><a href="#" onClick="print()"></a></p>
    <hr>
</div>
    <%}
    %>
</p>
<p align="center">&nbsp;</p>
</BODY>
</HTML>

```

nuevo_hardware.asp

```

<HTML>
<TITLE> Filtrar un registro </TITLE>
<%@ LANGUAGE=JScript %>
<!-- #INCLUDE File="ADOJAVAS.inc" -->
<SCRIPT LANGUAGE ="JavaScript">
<!-- se oculta la información de los navegadores antiguos

// final del comentario -->
</SCRIPT>
<BODY>
<%
    Ob_Conn = new ActiveXObject ("ADODB.Connection")
    Ob_RS=new ActiveXObject ("ADODB.RecordSet")
    Ob_Conn.Open ("Productos")
    Ob_RS.Open("hardware", Ob_Conn,adOpenStatic,adCmdTable)
    Ob_RS.AddNew()
    Ob_RS("nombre")= Request.Form("text_nombre")
    Ob_RS("dia_alta")= parseInt(Request.Form("text_dia_alta"))
    Ob_RS("mes_alta")= parseInt(Request.Form("text_mes_alta"))
    Ob_RS("ano_alta")= parseInt(Request.Form("text_ano_alta"))
    Ob_RS("descripcion")=Request.Form ("text_descripcion")

```

```

Ob_RS("controlbuap")=Request.Form ("text_controlbuap")
Ob_RS("ubicacion")=Request.Form ("text_ubicacion")
Ob_RS("estado")=Request.Form ("text_estado")
Ob_RS("dia_compra") =parseInt(Request.Form ("text_dia_compra"))
Ob_RS("mes_compra") =parseInt(Request.Form ("text_mes_compra"))
Ob_RS("ano_compra") =parseInt(Request.Form ("text_ano_compra"))
Ob_RS("costo_total") =Request.Form ("text_costo_total")
Ob_RS("no_factura") =Request.Form ("text_no_factura")
Ob_RS("organizacion_responsable") =Request.Form
("text_organizacion_responsable")
Ob_RS("unidad") =Request.Form ("text_unidad")
Ob_RS("endosar_a") =Request.Form ("text_endosar_a")
Ob_RS("como_se_adquirio") =Request.Form ("text_como_se_adquirio")
Ob_RS("marca") =Request.Form ("text_marca")
Ob_RS("modelo") =Request.Form ("text_modelo")
Ob_RS("no_serie") =Request.Form ("text_no_serie")
Ob_RS.Update()
Ob_Conn.Close()
%>
</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<table width="52%" border="1" align="center" cellpadding="1" cellspacing="1"
bordercolor="#CC9900" bgcolor="#990033">
  <tr>
    <td height="24"><div align="center"><strong><font color="#CC9900" size="4">Se
han dado de alta</font></strong></div></td>
  </tr>
</table>
<p>&nbsp;</p>

</BODY>
</HTML>

```

4. PRUEBAS

4.1 REQUERIMIENTOS DEL SISTEMA

Los requisitos para acceder al sistema creado son los siguientes:

Hardware:

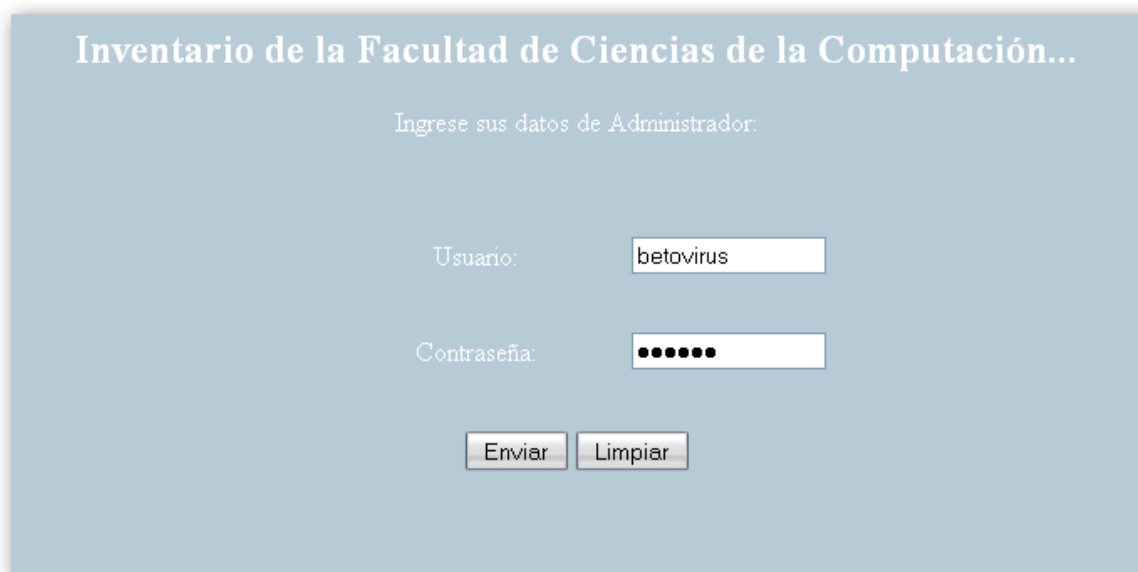
- Minimo de 128 mb RAM
- Espacio en disco duro de 4 GB.

Software:

- Conexión a Internet
- Navegador Web

4.2 INTERFAZ DEL USUARIO

4.2.1 Pantalla de Bienvenida



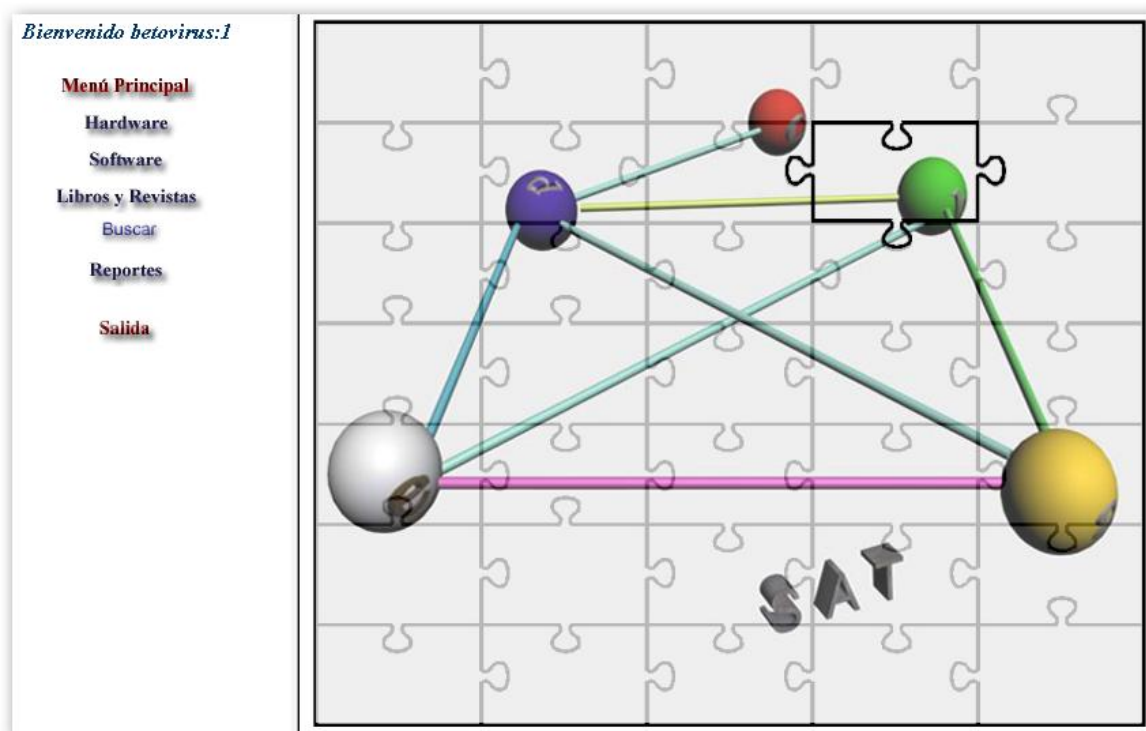
Inventario de la Facultad de Ciencias de la Computación...

Ingrese sus datos de Administrador:

Usuario:

Contraseña:

En esta ventana deberá autenticarse el administrador del sistema. En caso de lograr el acceso se mostrara la siguiente pantalla:



En caso contrario aparecerá la siguiente, donde se indica que no se le puede permitir el ingreso al usuario.

Lo sentimos, el nombre de usuario
y/o password es incorrecto! [volver](#)

Y regresará al usuario mediante el link a la pantalla inicial.

4.2.2 Menú principal



4.2.3 Pantalla de Hardware.

Nos despliega el material clasificado en hardware, paginado en grupos de 20 registros.

Hardware				
Página 1 DE 20				
NOMBRE	DESCRIPCION	CONTROL BUAP	UBICACION	
VIDEOPROYECTOR	CABLES, PLUG INS, MANUAL, ESTUCHE PORTACAÑON	PTRM094705	136202 LAB. MULTIMEDIA/LAB. PARALELISMO	<input type="radio"/>
COMPUTADORA PORTATIL	COMPUTADORA PORTATIL	PTRM038204	136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
CAMARA DIGITAL	CAMARA DIGITAL	PTRM032902	136202 LAB. MULTIMEDIA/LAB. PARALELISMO	<input type="radio"/>
-MONITOR	17 SONY TRINITRON MULTISCAN G220	461664-136202-2002	136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
DISCO SEAGATE			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
TECLADO MICROSOFT	MULTIMEDIA		136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
BOCINAS	PCWORKS		136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
MEMORIA RAM			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
-MONITOR	17 SONY TRINITRON MULTISCAN G220	461660-136202-2002	136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
DISCO SEAGATE			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
TECLADO MICROSOFT			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
BOCINAS	PCWORKS		136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
FLOPPY			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
MEMORIA RAM			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
-MONITOR	17 SONY TRINITRON MULTISCAN G220	461661-136202-2002	136202 LAB. MULTIMEDIA/LAB	<input checked="" type="radio"/>

Los botones realizan lo siguiente:

-Modificar: tras seleccionar un elemento de la lista, al elegir esta acción se mostrara un panel que permita modificar al elemento.


Modificar Datos de Hardware	
Nombre:	<input type="text" value="-MONITOR"/>
Descripción:	<input type="text" value="17 SONY TRINITRON MULTISCAN G220"/>
Control buap:	<input type="text" value="461661-136202-2002"/>
Ubicación:	<input type="text" value="136202 LAB. MULTIMEDIA/LAB"/>
Estado:	<input type="text" value="BUENAS CONDICIONES"/>
Fecha de Alta:	Día: <input type="text" value="2"/> Mes: <input type="text" value="2"/> Año: <input type="text" value="2007"/>
Etiqueta Activo:	<input type="text" value="0"/>
Fecha de Compra:	Día: <input type="text" value="1"/> Mes: <input type="text" value="7"/> Año: <input type="text" value="1971"/>
Costo Total:	<input type="text" value="0"/>
Número de Factura:	<input type="text" value="0"/>
Organización Responsable :	<input type="text" value="FACULTAD DE CS. DE LA COMPUTACION"/>
Unidad:	<input type="text" value="1"/>
Endosar a :	<input type="text" value="1"/>
Como se adquirio:	<input type="text" value="Compra"/>
Marca :	<input type="text" value="SONY"/>
Modelo:	<input type="text" value="TRINITRON"/>
No. serie :	<input type="text" value="MULTISCAN G220"/>
<input type="button" value="Enviar"/>	

-Nuevo: tras elegir esta opción, se mostrara un formulario solicitando los datos necesarios para realizar la alta de un nuevo material de hardware al inventario.

Ingresar Datos de Nuevo Hardware	
Nombre:	<input type="text"/>
Descripción:	<input type="text"/>
Control buap:	<input type="text"/>
Ubicación:	<input type="text"/>
Estado:	BUENAS CONDICIONES <input type="button" value="v"/>
Fecha de Alta:	Día: <input type="text"/> Mes: <input type="text"/> Año: <input type="text"/>
Etiqueta Activo:	<input type="text"/>
Fecha de Compra:	Día: <input type="text"/> Mes: <input type="text"/> Año: <input type="text"/>
Costo Total:	<input type="text"/>
Número de Factura:	<input type="text"/>
Organización Responsable :	<input type="text"/>
Unidad:	<input type="text"/>
Endosar a :	<input type="text"/>
Como se adquirió :	Compra <input type="button" value="v"/>
Marca :	<input type="text"/>
Modelo:	<input type="text"/>
No. serie :	<input type="text"/>
<input type="button" value="Enviar"/>	

-Eliminar: Borra del inventario al elemento seleccionado.

-Préstamo: Permite asignar en préstamo para algún docente el material seleccionado.

Hoja de Préstamo	
Clave de Control:	<input type="text" value="24"/>
Nombre:	-MONITOR
Descripción:	17 SONY TRINITRON MULTISCAN G220
Control buap:	461661-136202-2002
Marca :	SONY
Modelo:	TRINITRON
No. serie :	MULTISCAN G220
Fecha Prestamo:	Día: <input type="text" value="3"/> Mes: <input type="text" value="2"/> Año: <input type="text" value="2007"/>
Fecha Entrega:	Día: <input type="text" value="13"/> Mes: <input type="text" value="5"/> Año: <input type="text" value="2007"/>
Nombre	<input type="text" value="BAUTISTA RAMOS CESAR"/>
Firma	<hr/>
<input type="button" value="Confirmar"/>	
	

Dentro de este panel, se puede imprimir el formato de préstamo. Se deben de indicar fecha de préstamo y de entrega.

-Siguiente: Muestra los siguientes 20 registros del inventario.

4.2.4 Pantalla de Software

Nos despliega el material clasificado en hardware, paginado en grupos de 20 registros.

Software

TITULO	AREA	VERSION	ISBN	UBICACION	
JDK	COMPUTACION	1.5	1	BIBLIOTECA	<input type="radio"/>
VISUAL STUDIO	COMPUTACION	2009	1	MULTIMEDIA	<input type="radio"/>
HOLA MUNDO	MATEMATICAS	5.7	1	CAFETERIA	<input checked="" type="radio"/>

Los botones realizan lo siguiente:

-Modificar: tras seleccionar un elemento de la lista, al elegir esta acción se mostrara un panel que permita modificar al elemento.

Modificar Datos de Software

Clave de Profesor:

Area:

Autenticidad:

Titulo

Version:

Compañia

Copyrighth:

No. Serie

CD Key

No. Piezas

ISBN

Comentarios


Ubicacion

Nuevo: tras elegir esta opción, se mostrará un formulario solicitando los datos necesarios para realizar la alta de un nuevo material de software al inventario.

Agregar Nuevo Software	
Area:	<input type="text"/>
Autenticidad:	Original <input type="button" value="v"/>
Título	<input type="text"/>
Version:	<input type="text"/>
Copyright:	<input type="text"/>
Compañía	<input type="text"/>
No. Serie	<input type="text"/>
CD Key	<input type="text"/>
No. Piezas	<input type="text"/>
ISBN	<input type="text"/>
Comentarios	<input type="text"/>
Ubicacion	<input type="text"/>
<input type="button" value="Enviar"/>	

-Eliminar: Borra del inventario al elemento seleccionado.

-Préstamo: Permite asignar en préstamo para algún docente el material seleccionado.

Hoja de Prestamo	
Clave de Control:	<input type="text" value="66"/>
Título:	VISUAL STUDIO
Area:	COMPUTACION
Compañía:	MICROSOFT
Ubicación:	MULTIMEDIA
CD KEY:	78907979
Version :	2009
Fecha Prestamo:	Día: <input type="button" value="4"/> Mes: <input type="button" value="10"/> Año: <input type="button" value="2005"/>
Fecha Entrega:	Día: <input type="button" value="19"/> Mes: <input type="button" value="7"/> Año: <input type="button" value="1988"/>
Nombre	<input type="text" value="CASTRO CARDONA MAURICIO"/>
Firma	<hr/>
<input type="button" value="Confirmar"/>	
	

-Siguiente: Muestra los siguientes 20 registros del inventario.

4.2.5 Pantalla de Libros y Revistas.

Nos despliega el material clasificado en hardware, paginado en grupos de 20 registros.

Libros y Revistas				
Página 1 DE 4				
TITULO	UBICACION ACTUAL	AUTOR	EDITORIAL	
MANUAL DE WINDOWS 95	136202 LAB. MULTIMEDIA/LAB			
INTRODUCCION A WIDOWS 98	136202 LAB. MULTIMEDIA/LAB			
TIPOGRAFIA DECORATIVA	136202 LAB. MULTIMEDIA/LAB			
MANUAL DE DISEÑO GRAFICO	136202 LAB. MULTIMEDIA/LAB			
COMO CAMBIAR Y ELEGIR COLORES PARA DISEÑO GRAFICO	136202 LAB. MULTIMEDIA/LAB			
MANUAL DC1000 DVD100	136202 LAB. MULTIMEDIA/LAB			
ADOBE PREMIERE 6.0 USERS	136202 LAB. MULTIMEDIA/LAB			
FUNDAMENTOS DE LA TEORIA DE COLORES	136202 LAB. MULTIMEDIA/LAB			
OBTENGA RESULTADOS CON MICROSOFT OFFICE	136202 LAB. MULTIMEDIA/LAB			
ARCHIVOS MIDI	136202 LAB. MULTIMEDIA/LAB			
DE LA MULTIMEDIA A LA HIPERMEDIA	136202 LAB. MULTIMEDIA/LAB			
TODO SOBRE MULTIMEDIA	136202 LAB. MULTIMEDIA/LAB			
APRENDIENDO PROGRAMACION CON ACTIVEX EN 21 DIAS	136202 LAB. MULTIMEDIA/LAB			
COREL LUMINIERE	136202 LAB. MULTIMEDIA/LAB			
LIBRO PARA ANIMACION EN WEB	136202 LAB. MULTIMEDIA/LAB	9	9	

Modificar Nuevo Borrar Prestamo

Página Siguiente

Los botones realizan lo siguiente:

-Modificar: tras seleccionar un elemento de la lista, al elegir esta acción se mostrara un panel que permita modificar al elemento.

Modifica Libros y Revistas	
Titulo:	MANUAL DE WINDOWS 95
Ubicacion Actual :	136202 LAB. MULTIMEDIA/LAB
Estado:	BUENO
Fecha de Alta:	Día: 1 Mes: 1 Año: 1966
Etiqueta Activo:	879797
Costo Total:	900
Numero de Factura:	898645
Unidad:	PZA
Endosado a:	
Como se Adquirio:	COMPRA
Autor:	DAVIDSON
Editorial:	PATRIA
No. de Serie:	78979879879798797
Isbn, Issn:	HYU8797897
Descripcion:	BUENO
Tipo:	LIBRO
Enviar	

-Nuevo: tras elegir esta opción, se mostrará un formulario solicitando los datos necesarios para realizar la alta de un nuevo material de hardware al inventario.

Agrega Nuevo Libro o Revista	
Título:	<input type="text"/>
Ubicacion Actual :	<input type="text"/>
Estado Actual:	Bueno <input type="button" value="v"/>
Fecha de Alta:	Día: <input type="button" value="Día"/> <input type="button" value="v"/> Mes <input type="button" value="Mes"/> <input type="button" value="v"/> Año <input type="button" value="Año"/> <input type="button" value="v"/>
Etiqueta Activo:	<input type="text"/>
Costo Total:	<input type="text"/>
Numero de Factura:	<input type="text"/>
Catalogo Cuenta:	<input type="text"/>
Unidad:	<input type="text"/>
Endosado a:	<input type="text"/>
Como se Adquirio:	<input type="text"/>
Autor:	<input type="text"/>
Editorial:	<input type="text"/>
No. de Serie:	<input type="text"/>
Isbn, Issn:	<input type="text"/>
Descripcion:	<input type="text"/>
Tipo:	LIBRO <input type="button" value="v"/>
<input type="button" value="Enviar"/>	

-Eliminar: Borra del inventario al elemento seleccionado.

-Préstamo: Permite asignar en préstamo para algún docente el material seleccionado.

Hoja de Prestamo	
Clave de Control:	<input type="text" value="1"/>
Título:	MANUAL DE WINDOWS 95
Editorial:	PATRIA
Autor:	DAVIDSON
ISBN:	HYU8797897
Descripcion:	BUENO
Tipo :	1
Ubicacion Actual:	136202 LAB. MULTIMEDIA/LAB
Fecha Prestamo:	Día: <input type="button" value="2"/> <input type="button" value="v"/> Mes <input type="button" value="3"/> <input type="button" value="v"/> Año <input type="button" value="1968"/> <input type="button" value="v"/>
Fecha Entrega:	Día: <input type="button" value="4"/> <input type="button" value="v"/> Mes <input type="button" value="3"/> <input type="button" value="v"/> Año <input type="button" value="1969"/> <input type="button" value="v"/>
Nombre	<input type="text" value="CASTILLO ZACATELCO HILDA"/> <input type="button" value="v"/>
Firma	<hr/>
<input type="button" value="Confirmar"/>	
	

Dentro de este panel, se puede imprimir el formato de préstamo. Se deben de indicar fecha de préstamo y de entrega.

-Siguiente: Muestra los siguientes 20 registros del inventario.

4.2.6 Pantalla de Búsqueda de Material.

Nos permite buscar coincidencias en el nombre o la descripción de algún material con la cadena introducida por el usuario. Esta función implementada principalmente para encontrar algo muy específico o deseado por el usuario.

Por ejemplo: Introduciendo la palabra “cámara” se hayan los siguientes resultados:

Hardware				
NOMBRE	DESCRIPCION	CONTROL BUAP	UBICACION	
CAMARA DIGITAL	CAMARA DIGITAL	PTRM032902	136202 LAB. MULTIMEDIA/LAB. PARALELISMO	<input type="radio"/>
CAMARA VIVITAR UV-3300-SE	CAMARA 50 mmf/1.7 Lens & Case P/K 052 mm		136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
VIDEO CAMARA PANASONIC UVJ77 PALMORDER BATERIA CARGADOR Y ESTUCHE	COLOR LCD X22 DIGITAL ZOOM	211730	136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
VIDEOCAMARA SONY 8mm CARGADOR, BATERIA, ESTUCHE, CONTROL			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
CAMARA CANON ELPH10AF	CAMARA		136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
CAMARA DIGITAL CASIO QV 300 (con adaptador)		203432	136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
CAMARA DE VIDEO SONY	VIDEO Hi8 CON NIGHTSHOT(ADAPTADOR) AV CABLES		136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
TRIPIE CANNON PARA CAMARA			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
VIDEO CAMARA AVER MEDIA			136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
CAMARA DIGITAL CANON POWERSHOT S40	CABLE A/V USB PANTALLA LCD GUIA DE USUARIO	461931 136202 2003	136202 LAB. MULTIMEDIA/LAB	<input type="radio"/>
CAMARA DIGITAL CREATIVE PC-CAM 300	FLASH INTEGRADO GUIA DE USUARIO		136202 LAB. MULTIMEDIA/LAB	<input checked="" type="radio"/>

Y tal como se puede apreciar en la imagen, podemos realizar las mismas acciones que se manejan en las opciones principales de las categorías de material.

4.2.7 Pantalla de Reportes:

Dentro de este menú podemos filtrar la información por categoría, por ubicación, por estado, por área. Para cada reporte que se genera, al final se indica cuantos elementos cumplen la condición especificada.

REPORTES DE MATERIAL

Hardware

Por ubicacion

Por estado (SOLO
HARWARE)

Por Area (SOLO
SOFTWARE)

Material prestado a :

Por ejemplo: si se desea que hardware se encuentra en cierta ubicación, el procedimiento sería el siguiente:

REPORTES DE MATERIAL

Hardware

Por ubicacion

Por estado (SOLO
HARWARE)

Por Area (SOLO
SOFTWARE)

Material prestado a :

REPORTES DE MATERIAL

ELIJA LA UBICACION:

136202 LAB. MULTIMEDIA/LAB	Ver Reporte
136202 LAB. MULTIMEDIA/LAB	
136202 LAB. MULTIMEDIA/LAB. PARALELISMO	
BIBLIOTECA	

**Hardware en 136202 LAB.
MULTIMEDIA/LAB.
PARALELISMO**

NOMBRE	DESCRIPCION	UBICACION
VIDEOPROYECTOR	CABLES, PLUG INS, MANUAL, ESTUCHE PORTACAÑON	136202 LAB. MULTIMEDIA/LAB. PARALELISMO
CAMARA DIGITAL	CAMARA DIGITAL	136202 LAB. MULTIMEDIA/LAB. PARALELISMO

Total: 2

Además dentro de este menú podemos hacer consulta por profesor para saber que material ha adquirido en préstamo y poder llevar a cabo su devolución.

Por ejemplo, tomando al primer profesor, el reporte generado sería el siguiente:

HARDWARE

NOMBRE	DESCRIPCION	
MEMORIA RAM		<input type="radio"/>
-MONITOR	17 SONY TRINITRON MULTISCAN G220	<input checked="" type="radio"/>

Devolucion

SOFTWARE

TITULO	AREA	
		<input type="radio"/>

Devolucion

LIBROS Y REVISTAS

TITULO	AUTOR	
LIBRO PARA ANIMACION EN WEB	9	<input checked="" type="radio"/>

Devolucion

Total de Prestamos: 3

CONCLUSIONES

El objetivo primordial de la tesis era generar un sistema web que permitiera controlar la administración de préstamos de material en la Facultad de Ciencias de la Computación, así como permitir llevar a cabo reportes estadísticos sobre las existencias en inventario. Para ello, se tuvo que emplear conocimientos en bases de datos, desde su análisis hasta su diseño, aplicar técnicas de programación y utilizar todas aquellas herramientas para llevar a cabo la implementación.

La interfaz pretendí que fuera lo más agradable y comprensible para el usuario, sin perder de vista el garantizar la seguridad de la información.

Considero cubiertos los propósitos generales y me considero satisfecho con el esfuerzo realizado en conjunto con mi asesora.

REFERENCIAS

- [1] Date, C., “Introducción a los sistemas de bases de datos” , (Trad. Escalona,R.), (7ª ed.), México, Addison Wesley, 2001.
- [2] Miguel, A., Piattini, M., y Marcos, E., “Diseño de bases de datos relacionales” , México, Alfaomega, 2000.
- [3] Tim Martín. “Manual para programadores”, McGraw-Hill, 1991.
- [4] Date C.J. “Introducción a los Sistemas de bases de datos”, Addison-Wesley, 1993.
- [5] Connie U. Smith Performance Engineering of Software Systems. Addison Wesley. 1990
- [6] NAUR, P. Y B. RANDALL(eds.) Software Engineering: A Report on a Conference Sponsored by the NATO Science committee, NATO, 1969.
- [7]María del Rocío Boone Rojas, “Introducción a la Ingeniería de Software”, Diplomado en computación: Área de Bases de Datos, Otoño 2001.
- [8] Scot Hillier, Daniel Mezick. “Programación de Active Server Pages”. Ed. McGrawHill, Microsoft Press. ISBN: 84-481-1466-3.

INDICE

CAPITULO 1 – MARCO TEORICO

1.1 INTRODUCCIÓN.....	1
1.2 INGENIERÍA DE SOFTWARE	
1.2.1 Introducción a la Ingeniería de Software.....	1
1.2.2 Proceso, métodos y herramientas de la Ingeniería de Software	2
1.2.3 Características de calidad de un producto de programación.....	2
1.2.4 El ciclo de vida del software.....	2
1.3 BASES DE DATOS	
1.3.1 Conceptos básicos sobre bases de datos.....	3
1.3.2 Ventajas e inconvenientes de utilizar Bases de Datos.....	5
1.3.3 Que son los Sistemas Manejadores de Bases de Datos (DBMS)	7
1.3.4 Tareas y funciones generales del DBMS.....	8
1.3.5 El modelo de datos y sus cualidades.....	8
1.3.6 El Modelo Entidad – Relación (E-R)	9
1.3.7 Cualidades del modelo E-R.....	9
1.4 PAGINAS WEB DINAMICAS	
1.4.1 Introducción a ASP.....	10
1.4.2 Interacción entre el servidor y el navegador web cuando se utiliza ASP...	11
1.4.3 Otras alternativas.....	12
1.4.4 Herramientas necesarias	
1.4.4.1 El servidor web.....	12
1.4.4.2 El navegador.....	13
1.4.4.3 Herramientas para el desarrollo.....	13

CAPITULO 2 – ANALISIS Y DISEÑO

2.1 ANALISIS	
2.1.1 Definición del problema.....	14
2.2 ESPECIFICACION DE REQUERIMIENTOS	
2.2.1 Recopilación.....	14
2.2.2 Antecedentes.....	15
2.2.3 Alcances.....	15
2.2.4 Metodología.....	16
2.2.5 Restricciones.....	16
2.2.6 Requerimientos Del Sistema.....	17
2.2.7 Requerimientos Funcionales.....	17
2.2.8 Requerimientos No Funcionales.....	17
2.2.9 Descripción de la Información.....	18
2.3 DISEÑO	
2.3.1 Modelo Entidad - Relación.....	27
2.3.2 Modelo Relación de la Base de Datos.....	28
2.3.3 Diccionario de Datos.....	29

CAPITULO 3 – IMPLEMENTACION

3.1 INTRODUCCION	34
3.2 CONECTIVIDAD A LA BASE DE DATOS.....	34

3.2 DESARROLLO DEL SISTEMA	35
3.4 CODIGO FUENTE.....	36

CAPITULO 4 - PRUEBAS

4.1 REQUERIMIENTOS DEL SISTEMA	50
4.2 INTERFAZ DEL USUARIO	50
4.2.1 Pantalla de Bienvenida.....	50
4.2.2 Menú Principal.....	51
4.2.3 Pantalla de Hardware	52
4.2.4 Pantalla de Software.....	54
4.2.5 Pantalla de Libros y Revistas	56
4.2.6 Pantalla de Búsqueda de Material.....	58
4.2.7 Pantalla de Reportes	58
CONCLUSIONES	61
REFERENCIAS.....	62
INDICE	63