



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**DESARROLLO DE EXÁMENES EN LÍNEA
PARA LA CARRERA DE INGENIERÍA
MECÁNICA**

T E S I S

que para obtener el título de

INGENIERO EN COMPUTACIÓN

P R E S E N T A N

DANIEL VALENTIN LÓPEZ BENÍTEZ

LUIS ALBERTO REYES JUAREZ

DIRECTOR DE TESIS:

ING. MARIANO GARCÍA DEL GALLEGO





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

Agradezco a todas las personas que hicieron posible realizar este sueño brindándome su apoyo, cariño, confianza, compañía y orientación.

Daniel López

Dedico este trabajo a quienes me han apoyado y dado confianza, mi más profundo cariño, admiración y respeto a toda mi familia, maestros, amigos, y a todas aquellas personas que influyeron de una u otra forma en la realización de este sueño.

En especial a mi madre quien me formó con mucha dedicación, paciencia y cariño, fue mi mayor apoyo en todos los momentos importantes de mi vida, te agradezco infinitamente donde quiera que te encuentres...

A mi padre y hermana por su apoyo y paciencia, a quienes debo mucho lo que soy.

Gracias Luis Alberto.

Índice

INTRODUCCIÓN	1
1.- Redes de Computadoras	3
1.1 Tipos de Redes	3
1.2 Modelo OSI	4
1.3 El estándar TCP/IP	9
1.4 Qué es Internet	12
2.- Aplicaciones para Internet	15
2.1 HTML	15
2.1.1 Nociones básicas de HTML	16
2.1.2 Historia del estándar	18
2.2 VISUAL BASIC SCRIPT	19
2.3 ASP	20
2.4 SQL Server	21
2.4.1 ¿Qué son las bases de datos?	22
2.4.2 Normalización de una base de datos	25
2.4.3 Requisitos del Sistema	33
2.5 Página Dinámica	34
3.- Disponibilidad para Internet	38
3.1 Servidores	38
3.2 IIS (Internet Information Services)	39
3.2.1 Hardware	40
3.3 Apache	42
3.4 IIS VS APACHE	43
3.4.1 Fundamentos	44
3.4.2 Ambiente de ejecución	45
3.4.3 Componentes Dinámico	46
3.4.4 Seguridad y autenticación	48
3.4.5 Confiabilidad	49
4.- Diseño de exámenes en línea para la carrera de Ingeniería Mecánica	53
4.1 Cómo generar la base de datos	53
4.1.1 Consultas a la base de datos	58
4.2 Conexión de páginas ASP con una base de datos	61
4.3 Diseño de página dinámica	62
4.4 Implementación	65
4.5 Resultados del Diseño	73

5.- Proyecciones	76
5.1 Resultados esperados	76
5.2 Desarrollo propuesto	76
5.3 Seguridad	77
5.4 Manual de uso del sistema	79
6.- Conclusiones	81
Notas Bibliográficas	82
Bibliografía	83

Índice de tablas

Tabla 1.1 Niveles del modelo OSI.....	6
Tabla 1.2 Capas de TCP/IP	10
Tabla 2.1 Nivel de formalización cero	26
Tabla 2.2 Nivel de formalización 1	27
Tabla 2.3 Nivel de formalización 2	28
Tabla 2.4 Nivel de formalización 3	29
Tabla 2.5,2.6,2.7 Relación varios-varios	31
Tabla 3.1 Comparativa entre IIS y Apache	44
Tabla 3.2 Características de servidor de IIS y Apache	47
Tabla 3.3 Características de seguridad de IIS y Apache	49

Índice de figuras

Figura 1.0 Funcionamiento del ASP.....	20
Figura 4.0 Cómo entrar al ambiente SQL	54
Figura 4.1 Cómo generar la base de datos	55
Figura 4.2 Una tabla en la base de datos	56
Figura 4.3 Diagrama de la base de datos ingeprimer	57
Figura 4.4 Consulta de información	60
Figura 4.5 Consulta específica	61
Figura 4.6 Diagrama del sistema	64

Objetivo

Desarrollar un sistema de exámenes para aplicarse en la red (WEB).

INTRODUCCIÓN

Como resultado de que las tecnologías en la transferencia de datos se han desarrollado de forma tal que hoy en día es posible mantener comunicación casi directa con personas en otras partes del mundo por muy alejadas que estas se encuentren además de que es posible también mandar cualquier tipo de datos por la red internacional que es Internet, concretamente en el ámbito estudiantil esta red ha permitido que muchos estudiantes consulten la información que hay en esta y hagan aplicaciones dentro de ella con la finalidad de que el acceso a la información sea lo mas rápido y seguro posible por lo cual se ha pensado en desarrollar un sistema de exámenes en línea para así poder tener una referencia de la situación de los conocimientos adquiridos durante un curso antes de una evaluación general.

En este trabajo tratamos de abarcar los puntos más importantes en la elaboración de bases de datos y poder consultarlas por medio de Internet, esta es la idea fundamental de la tesis. La forma de acceder a la base de datos, cómo hacer consultas, además de poder ingresar y borrar información en ella, se realiza a través de páginas dinámicas con las cuales intentamos facilitar la tarea de aplicación de exámenes dentro de la carrera de Ingeniería Mecánica y posteriormente poder ofrecer el programa a las otras carreras.

Además, la tesis cuenta con temas introductorios que llevarán de la mano al lector para poder entender la elaboración de los exámenes en línea con las herramientas que nosotros utilizamos, y por qué utilizamos tales.

La tecnología utilizada en este proyecto nos es una tecnología generada, sin embargo logramos aplicar ésta para los fines necesarios al máximo gracias a la gran experiencia que hemos obtenido en la facultad durante nuestra estancia.

De esta forma este proyecto fue diseñado con las especificaciones dadas por los encargados Centro de Diseño y Manufactura, de la Facultad de Ingeniería. UNAM.



*REDES
DE
COMPUTADORAS*

1.- Redes de computadoras.

Las redes de computadoras han tenido un auge extraordinario en los últimos años y por medio de estas podemos intercambiar y compartir información entre diferentes usuarios a través del correo electrónico, crear grupos de discusión a distancia sobre diversos temas, tener acceso a bibliotecas electrónicas en lugares distantes, utilizar facilidades de cómputo en áreas geográficas diferentes y crear sistemas de procesamiento distribuido de transacciones, por mencionar algunas de las aplicaciones que actualmente se tienen.

Todos estos beneficios que se derivan de la utilización de las redes locales han sido posibles gracias a los avances logrados en el área de comunicación de datos.

Las redes computacionales que operan en la actualidad están formadas por una jerarquía de redes de área amplia, redes metropolitanas y redes locales interconectadas entre sí. Las redes que operan en áreas geográficas reducidas tales como un departamento, un edificio o una corporación son redes de área local. Algunas de estas redes están interconectadas entre sí formando redes metropolitanas y estas a su vez se interconectan a las redes de área amplia para permitir la comunicación entre puntos muy distantes geográficamente hablando. También se tienen redes de área local conectadas directamente a redes de área amplia.

Las redes de computadoras están hechas con enlaces de comunicaciones que transportan datos (sistema de comunicación), entre dispositivos conectados a la red.

Los enlaces (canales de comunicación) se pueden realizar con cables, fibras ópticas o cualquier otro medio de comunicación.

1.1. Tipos de redes.

- **Redes Locales:** Conocidas como LAN (Local Area Networks), son usadas para comunicar un conjunto de computadoras en un área geográfica pequeña, generalmente un edificio o un conjunto de edificios cercanos o en un campus.
-

- **Redes Metropolitanas:** También conocidas como MAN (Metropolitan Area Networks), cubren por lo general un área geográfica restringida a las dimensiones de una ciudad. Usualmente se componen de la interconexión de varias redes locales y utilizan alguna facilidad pública de comunicación de datos.
- **Redes de Área Amplia:** Las redes de área amplia, también denominadas WAN (Wide Area Networks), son las primeras redes de comunicación de datos que se utilizaron. Estas redes cubren áreas geográficas muy grandes, del tamaño de un país o incluso del mundo entero, como es el caso de la red Internet. (9)

1.2. Modelo OSI.

En 1977, la Organización Internacional de Estándares (OSI), integrada por industrias representativas del medio, creó un subcomité para desarrollar estándares de comunicación de datos que promovieran la accesibilidad universal y una interoperabilidad entre productos de diferentes fabricantes.

El resultado de estos esfuerzos es el Modelo de Referencia Interconexión de Sistemas Abiertos (ISO “International Standard Organization”).

El Modelo OSI es un lineamiento funcional para tareas de comunicaciones y, por consiguiente, no especifica un estándar de comunicación para dichas tareas. Sin embargo, muchos estándares y protocolos cumplen con los lineamientos del Modelo OSI.

Como se mencionó anteriormente, OSI nace de la necesidad de estandarizar los elementos que participan en la solución del problema de comunicación entre equipos de cómputo de diferentes fabricantes.

Estos equipos presentan diferencias en:

- *Procesador Central.*
 - *Velocidad.*
 - *Memoria.*
 - *Dispositivos de Almacenamiento.*
 - *Interfaces para Comunicaciones.*
 - *Códigos de caracteres.*
 - *Sistemas Operativos.*
-

Estructura del Modelo OSI de ISO

El objetivo perseguido por OSI establece una estructura que presenta las siguientes particularidades:

- **Estructura multinivel:** Se diseñó una estructura multinivel con la idea de que cada nivel se dedique a resolver una parte del problema de comunicación. Esto es, cada nivel ejecuta funciones específicas.
 - **El nivel superior utiliza los servicios de los niveles inferiores:** Cada nivel se comunica con su similar en otras computadoras, pero debe hacerlo enviando un mensaje a través de los niveles inferiores en la misma computadora. La comunicación internivel está bien definida. El nivel N utiliza los servicios del nivel N-1 y proporciona servicios al nivel N+1.
 - **Puntos de acceso:** Entre los diferentes niveles existen interfaces llamadas "puntos de acceso" a los servicios.
 - **Dependencias de Niveles:** Cada nivel es dependiente del nivel inferior y también del superior.
 - **Encabezados:** En cada nivel, se incorpora al mensaje un formato de control. Este elemento de control permite que un nivel en la computadora receptora se entere de que su similar en la computadora emisora está enviándole información. Cualquier nivel dado, puede incorporar un encabezado al mensaje. Por esta razón, se considera que un mensaje está constituido de dos partes: Encabezado e Información. Entonces, la incorporación de encabezados es necesaria aunque representa un lote extra de información, lo que implica que un mensaje corto pueda ser voluminoso.
-

Sin embargo, como la computadora destino retira los encabezados en orden inverso a como fueron incorporados en la computadora origen, finalmente el usuario sólo recibe el mensaje original.

En la tabla 1.1 se presentan los nombres de los 7 niveles del modelo OSI :

Tabla 1.1 Niveles del Modelo OSI.

Aplicación.
Presentación.
Sesión.
Transporte.
Red.
Enlace de datos.
Físico.

Nivel Físico: Define el medio de comunicación utilizado para la transferencia de información, dispone del control de este medio y especifica bits de control, mediante:

- Definir conexiones físicas entre computadoras.
- Describir el aspecto mecánico de la interfaz física.
- Describir el aspecto eléctrico de la interfaz física.
- Describir el aspecto funcional de la interfaz física.
- Definir la técnica de transmisión.
- Definir el tipo de transmisión.
- Definir la codificación de línea.
- Definir la velocidad de transmisión.
- Definir el modo de operación de la línea de datos.

Nivel Enlace de Datos: Este nivel proporciona facilidades para la transmisión de bloques de datos entre dos estaciones de red. Esto es, organiza los 1's y los 0's del nivel físico en formatos o grupos lógicos de información, con el fin de:

- Detectar errores en el nivel físico.
-

- Establecer esquema de detección de errores para las retransmisiones o reconfiguraciones de la red.
- Establecer el método de acceso que la computadora debe seguir para transmitir y recibir mensajes.
- Realizar la transferencia de datos a través del enlace físico.
- Enviar bloques de datos con el control necesario para la sincronía.

En general controla el nivel y es la interfaz con el nivel de red, al poder establecer una transmisión libre de errores.

Nivel de Red: Este nivel define el enrutamiento y el envío de paquetes entre redes.

- Es responsabilidad de este nivel establecer, mantener y terminar las conexiones.
- Este nivel proporciona el enrutamiento de mensajes, determinando si un mensaje en particular deberá enviarse al nivel 4 (Nivel de Transporte) o bien al nivel 2 (Enlace de datos).
- Este nivel conmuta, en ruta y controla la congestión de los paquetes de información en una sub-red.
- Define el estado de los mensajes que se envían a nodos de la red.

Nivel de Transporte: Este nivel actúa como un puente entre los tres niveles inferiores totalmente orientados a las comunicaciones y los tres niveles superiores totalmente orientados al procesamiento. Además, garantiza una entrega confiable de la información.

- Asegura que la llegada de datos del nivel de red encuentra las características de transmisión y calidad de servicio requerido por el nivel 5 (Sesión).
 - Este nivel define cómo direccionar la localidad física de los dispositivos de la red.
 - Asigna una dirección única de transporte a cada usuario.
 - Define una posible multicanalización. Esto es, puede soportar múltiples conexiones.
 - Define la manera de habilitar y deshabilitar las conexiones entre los nodos.
 - Determina el protocolo que garantiza el envío del mensaje.
-

- Establece la transparencia de datos así como la confiabilidad en la transferencia de información entre dos sistemas.

Nivel Sesión: Provee los servicios utilizados para la organización y sincronización del diálogo entre usuarios y el manejo e intercambio de datos.

- Establece el inicio y termino de la sesión.
- Recuperación de la sesión.
- Control del diálogo; establece el orden en que los mensajes deben fluir entre usuarios finales.
- Referencia a los dispositivos por nombre y no por dirección.
- Permite escribir programas que correrán en cualquier instalación de red.

Nivel Presentación: Traduce el formato y asignan una sintaxis a los datos para su transmisión en la red.

- Determina la forma de presentación de los datos sin preocuparse de su significado o semántica.
- Establece independencia a los procesos de aplicación considerando las diferencias en la representación de datos.
- Proporciona servicios para el nivel de aplicaciones al interpretar el significado de los datos intercambiados.
- Opera el intercambio.
- Opera la visualización.

Nivel Aplicación: Proporciona servicios al usuario del Modelo OSI.

- Proporciona comunicación entre dos procesos de aplicación, tales como: programas de aplicación, aplicaciones de red, etc.
- Proporciona aspectos de comunicaciones para aplicaciones específicas entre usuarios de redes: manejo de la red, protocolos de transferencias de archivos (ftp), etc.

1.3. Estandar TCP/IP.

Se han desarrollado diferentes familias de protocolos para comunicación por red de datos para los sistemas de software libre y

software con licencia. El protocolo utilizado con mayor frecuencia es el Internet Protocol Suite, comúnmente conocido como TCP / IP.

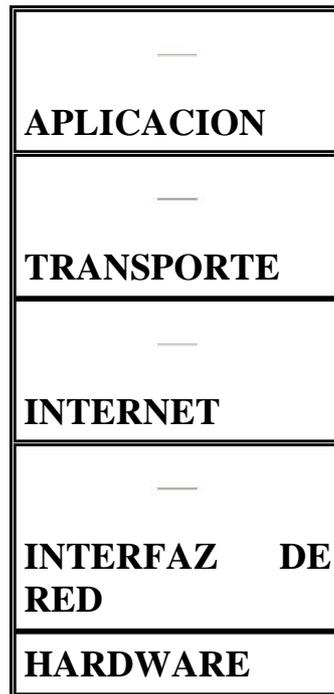
Definición. “ El nombre TCP / IP proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). En conjunto llegan a ser más de 100 protocolos diferentes definidos en éste”.(1)

El TCP / IP es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes LAN y redes WAN. TCP / IP fue desarrollado y demostrado por primera vez en 1972 por el departamento de defensa de los Estados Unidos, ejecutándolo en el ARPANET una red de área extensa del departamento de defensa.

Capas TCP/IP de internet

El segundo modelo mayor de estratificación por capas no se origina de un comité de estándares, sino que proviene de las investigaciones que se realizan respecto al conjunto de protocolos de TCP/IP. Con un poco de esfuerzo, el modelo ISO puede ampliarse y describir el esquema de estratificación por capas del TCP/IP, pero los presupuestos subyacentes son lo suficientemente distintos para distinguirlos como dos diferentes.

En términos generales, el software TCP/IP está organizado en cuatro capas conceptuales que se construyen sobre una quinta capa de hardware. La tabla 1.2 muestra las capas conceptuales del TCP/IP.

Tabal 1.2 Capas de TCP/IP**Descripción de las capas del protocolo TCP/IP**

Capa de aplicación. Es el nivel más alto, los usuarios llaman a una aplicación que acceda servicios disponibles a través de la red de redes TCP/IP. Una aplicación interactúa con uno de los protocolos de nivel de transporte para enviar o recibir datos. Cada programa de aplicación selecciona el tipo de transporte necesario, el cual puede ser una secuencia de mensajes individuales o un flujo continuo de octetos. El programa de aplicación pasa los datos en la forma requerida hacia el nivel de transporte para su entrega.

Capa de transporte. La principal tarea de la capa de transporte es proporcionar la comunicación entre un programa de aplicación y otro. Este tipo de comunicación se conoce frecuentemente como comunicación punto a punto. La capa de transporte regula el flujo de información. Puede también proporcionar un transporte confiable, asegurando que los datos lleguen sin errores y en secuencia. Para hacer esto, el software de protocolo de transporte tiene el lado de recepción enviando acuses de recibo de retorno y la parte de envío retransmitiendo los paquetes perdidos. El

software de transporte divide el flujo de datos que se está enviando en pequeños fragmentos (por lo general conocidos como paquetes) y pasa cada paquete, con una dirección de destino, hacia la siguiente capa de transmisión.

Aun cuando en el esquema anterior se utiliza un solo bloque para representar la capa de aplicación, una computadora de propósito general puede tener varios programas de aplicación accedendo la red de redes al mismo tiempo. La capa de transporte debe aceptar datos desde varios programas de usuario y enviarlos a la capa del siguiente nivel. Para hacer esto, se añade información adicional a cada paquete, incluyendo códigos que identifican qué programa de aplicación envía y qué programa debe recibir, así como una suma de verificación para verificar que el paquete ha llegado intacto y utiliza el código de destino para identificar el programa de aplicación en el que se debe entregar.

Capa Internet. La capa Internet maneja la comunicación de una máquina a otra. Ésta acepta una solicitud para enviar un paquete desde la capa de transporte, junto con una identificación de la máquina, hacia la que se debe enviar el paquete.

La capa Internet también maneja la entrada de datagramas, verifica su validez y utiliza un algoritmo de ruteo para decidir si el datagrama debe procesarse de manera local o debe ser transmitido. Para el caso de los datagramas direccionados hacia la máquina local, el software de la capa de red de redes borra el encabezado del datagrama y selecciona, de entre varios protocolos de transporte, un protocolo con el que manejará el paquete. Por último, la capa Internet envía los mensajes ICMP(poner el nombre de las iniciales) de error y control necesarios y maneja todos los mensajes ICMP entrantes.

Capa de interfaz de red. El software TCP/IP de nivel inferior consta de una capa de interfaz de red responsable de aceptar los datagramas IP y transmitirlos hacia una red específica. Una interfaz de red puede consistir en un dispositivo controlador (por ejemplo, cuando la red es una red de área local a la que las máquinas están conectadas directamente) o un complejo subsistema que utiliza un protocolo de enlace de datos propios (por ejemplo, cuando la red consiste de conmutadores de paquetes que se comunican con anfitriones utilizando HDLC (que son las iniciales)).

1.4. Que es internet.

“ Internet es una red de computadoras que utiliza convenciones comunes a la hora de nombrar y direccionar sistemas. Es una colección de redes independientes interconectadas; no hay nadie que sea dueño o active Internet completamente”.(2)

Las computadoras que componen Internet trabajan en UNIX, el sistema operativo Macintosh, Windows y muchos otros. Utilizando TCP/IP y los protocolos veremos dos servicios de red:

- Servicios de Internet a nivel de aplicación
- Servicios de Internet a nivel de red

Servicios de Internet a nivel de aplicación:

Desde el punto de vista de un usuario, una red de redes TCP/IP aparece como un grupo de programas de aplicación que utilizan la red para llevar a cabo tareas útiles de comunicación. Utilizamos el término interoperabilidad para referirnos a la habilidad que tienen diversos sistemas de computación para cooperar en la resolución de problemas computacionales. Los programas de aplicación de Internet muestran un alto grado de interoperabilidad. Los servicios de aplicación de Internet más populares y difundidos incluyen:

Correo electrónico. Éste permite que un usuario escriba mensajes (texto, imágenes, audio, video, etc.) y los envíe a individuos o grupos. Otra parte de la aplicación de correo permite que un usuario lea los mensajes que ha recibido. El correo electrónico ha sido tan exitoso que muchos usuarios de Internet dependen de él para su correspondencia normal y de negocios. Aunque existen muchos sistemas de correo electrónico, al utilizar TCP/IP se logra que la entrega sea más confiable debido a que no se basa en ¿compradoras? intermedias para distribuir los mensajes de correo. Un sistema de entrega de correo TCP/IP opera al hacer que la máquina del transmisor contacte directamente la máquina del receptor. Por lo tanto, el transmisor sabe que, una vez que el mensaje salga de su máquina local, se habrá recibido de manera exitosa en el sitio de destino.

Transferencia de archivos. Aunque los usuarios algunas veces transfieren archivos por medio del correo electrónico, el correo está diseñado principalmente para mensajes cortos de texto. Los protocolos TCP/IP incluyen un programa de aplicación para transferencia de archivos, el cual permite que lo usuarios envíen o reciban archivos arbitrariamente grandes de programas o de datos. Por ejemplo, al utilizar el programa de transferencia de archivos, se puede copiar de una máquina a otra una gran base de datos que contenga imágenes de satélite, un programa escrito en Pascal o C++, o un diccionario del idioma inglés. El sistema proporciona una manera de verificar que los usuarios cuenten con autorización o, incluso, de impedir el acceso. Como el correo, la transferencia de archivos a través de una red de redes TCP/IP es confiable debido a que las dos máquinas comprendidas se comunican de manera directa, sin tener que confiar en máquinas intermedias para hacer copias del archivo a lo largo del camino.

Acceso remoto. El acceso remoto permite que un usuario que esté frente a una computadora se conecte a una máquina remota y establezca una sesión interactiva. El acceso remoto hace aparecer una ventana en la pantalla del usuario, la cual se conecta directamente con la máquina remota al enviar cada golpe de tecla desde el teclado del usuario a una máquina remota y muestra en la ventana del usuario cada carácter que la computadora remota lo genere. Cuando termina la sesión de acceso remoto, la aplicación regresa al usuario a su sistema local.

*APLICACIONES
PARA
INTERNET*

2.- APLICACIONES PARA INTERNET

Las aplicaciones más conocidas o populares de Internet son:

- **Web - www (World Wide Web)**

Sistema mundial de información; sencillo pero poderoso que puede mostrar texto, imágenes, audio y video, animaciones, etc.

- **Email - Correo Electrónico (Electronic Mail)**

Para enviar y recibir mensajes electrónicos en segundos a nivel mundial.

Otras aplicaciones de gran utilidad son:

- **Transferencia de Archivos FTP - (File Transfer Protocol)**

Sirve para transferir archivos entre usuarios y computadoras en todo el mundo.

- **Newsgroups - Foros de Discusión**

Es un mundo de información y debate mejor conocido como USENET

- **Chat - IRC (Internet Relay Chat)**

Para platicar en vivo con otras personas por medio de texto.

- **Telnet - Computación a distancia**

Aplicación para personas que usan computadoras remotas para diversos fines.

2.1. HTML

“El HTML, acrónimo inglés de HyperText Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato

estándar de las páginas web”(3). Gracias a Internet y a los navegadores del tipo Explorer, Mozilla, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTML es hijo de SGML(Standard Generalized Markup Language), aunque hay unas versiones de XHTML que son descendientes de XML(eXtensible Markup Language) y exigen que se escriba mucho más para facilitar la vida a los navegadores, que son aquellos que nos muestran información en pantalla.

2.1.1 Nociones básicas de HTML

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que deben aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador. Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (<>), y algunas tienen atributos que pueden tomar algún valor. En general las etiquetas se aplicarán de dos formas especiales:

Se abren y se cierran, como por ejemplo: negrilla que se vería en su navegador como el texto negrilla en negrilla.

No pueden abrirse y cerrarse, como <hr> que se vería en su navegador como una línea horizontal.

Otras que pueden abrirse y cerrarse, como por ejemplo <p>.

Las etiquetas básicas o mínimas son:

```
<html>
```

```
<head>
```

```
<title>Ejemplo</title>
```

```
</head>
```

```
<body>
```

ejemplo

```
</body>
```

```
</html>
```

Las etiquetas básicas de HTML, de obligada presencia en todo documento son:

<HTML>: Es la etiqueta que define el inicio del documento html, le indica al navegador que todo lo que viene a continuación debe tratarlo como una serie de códigos html.

<HEAD>: Define la cabecera del documento html, esta cabecera suele contener información sobre el documento que no se muestra directamente en el navegador. Como por ejemplo el título de la ventana de su navegador. Dentro de la cabecera **<HEAD>** podemos encontrar:

<TITLE>: Define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana

<LINK>: para definir algunas características avanzadas, como por ejemplo las stylesheets (hojas de estilo) usadas para el diseño de la página, ejemplo:
:<link rel="stylesheet" href="/style.css" type="text/css">

<BODY>: Define el contenido principal o cuerpo del documento, esta es la parte del documento html que se muestra en el navegador, dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes. Dentro del cuerpo **<BODY>** podemos encontrar numerosísimas etiquetas. A continuación se indican algunas a modo de ejemplo:

<H1>, **<H2>**,... **<H6>**: encabezados o títulos del documento en diferentes tamaños de fuente

<P>: párrafo nuevo

**
**: salto de línea forzado

<TABLE>: comienzo de una tabla (las filas se identifican con <TR> y las celdas dentro de las filas con <TD>)

<A>: indica la existencia de un hipervínculo o enlace, dentro o fuera la página web. Debe definirse el parámetro de pasada por medio del href

<DIV>: comienzo de un área especial en la página

2.1.2 Historia del estándar

No hay especificación oficial del HTML 1.0 porque ya existían múltiples estándares informales del HTML cuando se decidió crear un estándar oficial. Los trabajos para crear un sucesor del HTML, posteriormente llamado 'HTML+', comenzaron a finales de 1993. El HTML+ se diseñó originalmente para ser un superconjunto del HTML que permitiera evolucionar gradualmente desde el formato HTML anterior. A la primera especificación formal de HTML+ se le dio, por lo tanto, el número de versión 2.0 para distinguirla de esos "estándares no oficiales" previos. Los trabajos sobre HTML+ continuaron, pero nunca se convirtió en un estándar.

El borrador del estándar HTML 3.0 fue propuesto por el recién formado W3C (World Wide Web Consortium) en marzo de 1995. Con él se introdujeron muchas nuevas capacidades, tales como facilidades para crear tablas, hacer que el texto fluyese alrededor de las figuras y mostrar elementos matemáticos complejos. Aunque se diseñó para ser compatible con HTML 2.0, era demasiado complejo para ser implementado con la tecnología de la época y, cuando el borrador del estándar expiró en septiembre de 1995, se abandonó debido a la carencia de apoyos de los fabricantes de navegadores web. El HTML 3.1 nunca llegó a ser propuesto oficialmente, y el estándar siguiente fue el HTML 3.2, que abandonaba la mayoría de las nuevas características del HTML 3.0 y, a cambio, adoptaba muchos elementos desarrollados inicialmente por los navegadores web Netscape y Mosaic

El HTML 4.0 también adoptó muchos elementos específicos desarrollados inicialmente para un navegador web concreto, pero al mismo

tiempo comenzó a limpiar el HTML señalando algunos de ellos como “desaprobados”.

Ya no va a haber nuevas versiones del HTML. Sin embargo, la herencia del HTML se mantiene en XHTML, que se basa en XML.

2.2 Visual Basic Script

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de JavaScript.

Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows. Tanto su sintaxis como la manera de trabajar están muy inspirados en él. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero.

El modo de funcionamiento de Visual Basic Script para construir efectos especiales en páginas web es muy similar al utilizado en JavaScript y los recursos a los que se puede acceder también son los mismos: el navegador.

Como decimos, no debemos utilizar este lenguaje en la mayoría de las ocasiones, aunque un caso donde tendría sentido utilizar Visual Basic Script sería la construcción de una Intranet donde sepamos con toda seguridad que los navegadores que se van a conectar serán siempre Internet Explorer. En este caso, un programador habitual de Visual Basic tendría más facilidades para realizar los scripts utilizando Visual Basic Script en lugar de JavaScript.

2.3 ASP

Microsoft introdujo esta tecnología llamada **Active Server Pages(ASP)** en diciembre de 1996, por lo que no es nada nueva. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente.

Traduciendo la definición de **Microsoft**: "*Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web*".(4)

El principio de la tecnología ASP es el **VBScript**, pero existe otra diversidad de lenguajes de programación que pueden ser utilizados como lo es Perl, JScript, etc. El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador.

Para explicar mejor el funcionamiento del ASP se presenta la siguiente gráfica:

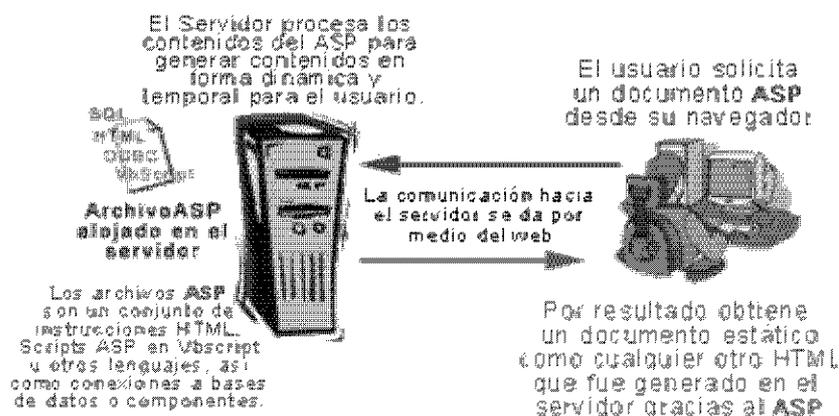


Fig 1.0 Funcionamiento del ASP

El desarrollo que se ha venido dando a lo que es ASP ha sido bastante amplio. Entre sus funciones principales están el acceso a base de datos, envío de correo electrónico, creación dinámica de gráficos y otros. Básicamente, muchas cosas que podemos realizar por medio de CGI (**Common Gateway Interface**) pueden ser realizadas con esta tecnología. Esto debido a que el ASP es tan eficiente con escribir código directamente a la interfase de aplicación del servidor, con la ventaja de que es más eficiente que el CGI que depende de un compilador ya que el ASP corre como un servicio en el servidor, tomando ventaja de la arquitectura de multitareas.

Para empezar con el desarrollo de las **Active Server Pages** es necesario un servidor con Windows NT 4.x o mayor y el Internet Information Server (IIS). El IIS es una aplicación gratuita que puede conseguirse en el Option Pack del NT. También es posible utilizar ASP en Windows 9x por medio del Personal Web Server junto al Option Pack mencionado anteriormente.

En caso del uso de un servidor Linux, Chilisoft ha desarrollado el *Chilisoft ASP* que también permite el uso de esta tecnología. Otra opción es *InstantAsp* también para correr ASP en servidores que no sean el IIS.

2.4 SQL server

Dado que la mayor parte de las aplicaciones, por no decir todas, deben almacenar, recuperar y tratar datos, un factor importante en el diseño de una infraestructura de aplicaciones es la forma en que éstos se administran. Microsoft® SQL Server™ 2000 proporciona servicios seguros y escalables de almacenamiento de datos relacionales, incluye completas capacidades de administración y permite obtener una gran disponibilidad. Sin embargo, aunque SQL Server está diseñado para "optimizarse automáticamente" todo lo posible, en toda aplicación deben tenerse en cuenta algunas consideraciones para cumplir los requisitos específicos de seguridad, rendimiento, escalabilidad y disponibilidad de la solución de Microsoft Internet Data Center. Este capítulo está orientado a identificar dichas consideraciones y discutir estrategias para diseñar una solución basada en SQL Server para la arquitectura de Internet Data Center.

Microsoft SQL Server "...es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a

disposición de muchos usuarios grandes cantidades de datos de manera simultánea..”(5).

Sus características son:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL (Data Definition Lenguaje) y DML(Data Manipulation Lenguaje) gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accesan a la información.
- Además permite administrar información de otros servidores de datos

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET.

Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

2.4.1 ¿Qué son las bases de datos?

“Las bases de datos son almacenes de información. Nada más. Desde un simple archivo de texto hasta un complejo sistema de inventarios que ocupe varios equipos”. (10)

“Las bases de datos son uno de los aspectos más importantes de la computación, lo cual explica su importancia en el desarrollo para Web”.(6)

Las bases de datos son una forma eficiente de guardar grandes cantidades de datos. Pero administrar tal información repercute en el rendimiento. Aunque las bases de datos son extremadamente eficientes para almacenar grandes cantidades de datos complejos relacionados entre sí, tal vez sólo necesite utilizar una cookie o una variable de sesión si desea

almacenar información temporal o en muy pequeña cantidad como el nombre de un visitante o su correo electrónico.

En la actualidad, y en gran parte gracias a la tecnología y recursos disponibles provenientes de campos como la informática y la electrónica, las bases de datos pueden adquirir diversas formas, ofreciendo un amplio rango de soluciones al problema de almacenar datos.

Las bases de datos pueden dividirse en dos grupos, considerando su función primordial, a saber:

Bases de datos analíticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

Bases de datos dinámicas

Éstas son bases de datos más dinámicas, orientadas a almacenar información que es modificada con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consultas

Modelos de bases de datos

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema

eficiente de *base de datos*; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

Bases de datos jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se les conoce como *hojas*.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

Bases de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Bases de datos relacionales

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por *registros* (las filas de una tabla), que representarían las tuplas, y *campos* (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

2.4.2 Normalización de una base de datos

Uno de los factores más importantes en la creación de páginas web dinámicas es el diseño de las Bases de Datos (BD). Si las tablas de la base de datos no están correctamente diseñadas, pueden causar un montón de dolores de cabeza cuando se realicen complicadísimas consultas SQL para extraer los datos que necesitas. Es necesario estudiar la normalización para con ella poder establecer las relaciones entre los datos. Una vez entendido este concepto podemos proseguir con el desarrollo de aplicaciones.

Básicamente, las reglas de Normalización están encaminadas a eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas. Más adelante explicaremos lo que esto significa mientras vemos los cinco pasos progresivos para normalizar. Debemos tomar en cuenta que al crear una BD ésta debe ser funcional y eficiente. También debemos tomar en cuenta a los tipos de relaciones que la estructura de datos pueda tener.

Digamos que queremos crear una tabla con la información de usuarios, y los datos a guardar son el nombre, la empresa, la dirección de la empresa y algún e-mail, o bien URL si las tienen. En principio comenzaremos definiendo la estructura de una tabla 2.1 como esta:

Formalización CERO

Tabla 2.1 Nivel de formalización cero

Usuarios				
Nombre	empresa	direccion_empresa	url1	url2
Joe	ABC	1 Work Lane	abc.com	xyz.com
Hill	XYZ	1 Job Street	abc.com	xyz.com

La tabla anterior está situada en el nivel de Formalización Cero porque ninguna de nuestras reglas de normalización ha sido aplicada. Observemos los campos url1 y url2 -- ¿Qué haremos cuando en nuestra aplicación necesitemos una tercera url ? ¿ Queremos tener que añadir otro campo/columna a tu tabla y tener que reprogramar toda la entrada de datos de tu código ? Obviamente no, lo que necesitamos crear es un sistema funcional que pueda crecer y adaptarse fácilmente a los nuevos requisitos. Hechemos un vistazo a las reglas del Primer Nivel de Formalización-Normalización, y apliquémoslas a nuestra tabla.

Primer nivel de Formalización/Normalización. (F/N)

1. Eliminar los grupos repetitivos de la tablas individuales.
2. Crear una tabla separada por cada grupo de datos relacionados.
3. Identificar cada grupo de datos relacionados con una clave primaria.

Una vez que aplicamos el primer nivel de F/N obtenemos la siguiente tabla 2.2:

Tabla 2. 2 Primer nivel de formalización

Usuarios				
UserId	nombre	Empresa	direccion_empresa	url
1	Joe	ABC	1 Work Lane	abc.com
1	Joe	ABC	1 Work Lane	xyz.com
2	Jill	XYZ	1 Job Street	abc.com
2	Jill	XYZ	1 Job Street	xyz.com

Ahora diremos que nuestra tabla está en el primer nivel de F/N. Hemos solucionado el problema de la limitación del campo url. Pero sin embargo vemos otros problemas....Cada vez que introducimos un nuevo registro en la **tabla usuarios**, tenemos que duplicar el nombre de la empresa y del usuario. No sólo nuestra BD crecerá muchísimo, sino que será muy facil que la BD se corrompa si escribimos mal alguno de los datos redundantes. Apliquemos entonces el segundo nivel de F/N:

Segundo nivel de F/N

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar estas tablas mediante una clave externa.

Hemos separado el campo url en otra tabla 2.3, de forma que podemos añadir más en el futuro si tener que duplicar los demás datos. También vamos a usar nuestra clave primaria para relacionar estos campos:

Tabla 2. 3 Segundo nivel de formalización

Usuarios			
userId	nombre	Empresa	direccion_empresa
1	Joe	ABC	1 Work Lane
2	Jill	XYZ	1 Job Street

Urls		
urlId	relUserId	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

Ahora la tabla se ha convertido en tablas separadas y la clave primaria en la **tabla usuarios**, userId, esta relacionada ahora con la clave externa en la **tabla urls**, relUserId. Esto esta mejor. ¿ Pero qué ocurre cuando queremos añadir otro empleado a la

empresa ABC ? ¿ o 200 empleados ? Ahora tenemos el nombre de la empresa y su dirección duplicandose, otra situación que puede inducirnos a introducir errores en nuestros datos. Así que tendremos que aplicar el tercer nivel de F/N:

Tercer nivel de F/N

1. Eliminar aquellos campos que no dependan de la clave.

Nuestro nombre de empresa y su dirección no tienen nada que ver con el campo `userId`, así que tienen que tener su propio `empresaId` en la tabla 2. 4:

Tabla 2. 4 Tercer nivel de formalización

Usuarios		
<code>userId</code>	Nombre	<code>relEmpresaId</code>
1	Joe	1
2	Hill	2

empresas		
<code>emprId</code>	Empresa	<code>direccion_empresa</code>
1	ABC	1 Work Lane
2	XYZ	1 Job Street

Urls		
<code>urlId</code>	<code>RelUserId</code>	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

Ahora tenemos la clave primaria `emprId` en la **tabla empresas** relacionada con la clave externa `recEmpresaId` en la **tabla usuarios**, y

podemos añadir 200 usuarios mientras que sólo tenemos que insertar el nombre 'ABC' una vez. Nuestras tablas de usuarios y urls pueden crecer todo lo que quieran sin duplicación ni corrupción de datos. La mayoría de los desarrolladores dicen que el tercer nivel de F/N es suficiente, que nuestro esquema de datos puede manejar fácilmente los datos obtenidos de cualquier empresa en su totalidad, y en la mayoría de los casos esto será cierto.

Pero echemos un vistazo a nuestro campo urls - ¿Notan duplicación de datos ? Esto es perfectamente aceptable si la entrada de datos de este campo es solicitada al usuario en nuestra ampliación para que teclee libremente su url, y por lo tanto es sólo una coincidencia que Joe y Jill teclearon la misma url. ¿ Pero que pasa si en lugar de entrada libre de texto usáramos un menú desplegable con 20 o incluso más urls predefinidas ? Entonces tendríamos que llevar nuestro diseño de BD al siguiente nivel de F/N, el cuarto, muchos desarrolladores lo pasan por alto porque depende mucho de un tipo muy específico de relación, la relación “varios-con-varios”, la cual aún no hemos encontrado en nuestra aplicación.

Relaciones entre los Datos

Antes de definir el cuarto nivel de F/N, veremos tres tipos de relaciones entre los datos: uno-a-uno, uno-con-varios y varios-con-varios. Mira la **tabla usuarios** en el Primer Nivel de F/N del ejemplo de arriba. Por un momento imaginemos que ponemos el campo url en una tabla separada, y cada vez que introducimos un registro en la **tabla usuarios** también introducimos una sola fila en la **tabla urls**. Entonces tendríamos una relación uno-a-uno: cada fila en la tabla usuarios tendría exactamente una fila correspondiente en la **tabla urls**. Para los propósitos de nuestra aplicación no sería útil la normalización.

Ahora observemos las tablas en el ejemplo del Segundo Nivel de F/N. Nuestras tablas permiten a un sólo usuario tener asociadas varias urls. Esta es una relación uno-con-varios, el tipo de relación más común, y hasta que se nos presentó el dilema del Tercer Nivel de F/N. la única clase de relación que necesitamos para este ejemplo.

La relación varios-con-varios, sin embargo, es ligeramente más compleja. Observemos que en nuestro ejemplo del Tercer Nivel de F/N tenemos a un usuario relacionado con varias urls. Como dijimos, vamos a cambiar la estructura para permitir que varios usuarios estén relacionados con varias urls y así tendremos una relación varios-con-varios. Veamos como quedarían nuestras tablas 2.5, 2.6, 2.7 antes de seguir con este planteamiento:

Relación varios-varios

Tabla 2.5

Usuarios		
userId	Nombre	relEmpresaId
1	Joe	1
2	Hill	2

Tabla 2.6

Urls	
urlId	url
1	abc.com
2	xyz.com

Tabla 2.7

url_relations		
relationId	relatedUrlId	relatedUserId
1	1	1
2	1	2
3	2	1
4	2	2

Empresas		
emprId	Empresa	direccion_empresa
1	ABC	1 Work Lane
2	XYZ	1 Job Street

Para disminuir la duplicación de los datos (este proceso nos llevará al Cuarto Nivel de F/N), hemos creado una tabla que sólo tiene claves externas y primarias `url_relations`. Hemos sido capaces de remover la entradas duplicadas en la tabla `urls` creando la tabla `url_relations`. Ahora podemos expresar fielmente la relación que ambos Joe and Jill tienen entre cada uno de ellos, y entre ambos, las urls. Así que veamos exactamente que es lo que el Cuarto Nivel de F/N. supone:

Cuarto Nivel de F/N.

1. En las relaciones varios-con-varios, entidades independientes no pueden ser almacenadas en la misma tabla.

Ya que sólo se aplica a las relaciones varios-con-varios, la mayoría de los desarrolladores pueden ignorar esta regla de forma correcta. Pero es muy útil en ciertas situaciones, tal como esta. Hemos optimizado nuestra **tabla urls** eliminado duplicados y hemos puesto las relaciones en su propia tabla.

A continuación se presenta un ejemplo práctico. Ahora podemos seleccionar todas las urls de Joe realizando la siguiente instrucción SQL:

```
SELECT nombre, url FROM usuarios, urls, url_relations WHERE
url_relations.relatedUserId = 1 AND usuarios.userId = 1 AND
urls.urlId = url_relations.relatedUrlId
```

Y si queremos recorrer todas las urls de cada uno de los usuarios, haríamos algo así:

```
SELECT nombre, url FROM usuarios, urls, url_relations WHERE
usuarios.userId = url_relations.relatedUserId AND urls.urlId =
url_relations.relatedUrlId
```

Quinto Nivel de F/N.

Existe otro nivel de normalización que se aplica a veces, pero es de hecho algo esotérico y en la mayoría de los casos no es necesario para obtener la mejor funcionalidad de nuestra estructura de datos o aplicación. Su principio sugiere:

1. La tabla original debe ser reconstruida desde las tablas resultantes en las cuales a sido troceada.

Los beneficios de aplicar esta regla aseguran que no has creado ninguna columna extraña en tus tablas y que la estructura de las tablas que has creado sea del tamaño justo que tiene que ser. Es una buena práctica aplicar esta regla, pero a no ser que estemos tratando con una extensa estructura de datos probablemente no sea necesaria.

Las tres primeras reglas de normalización fueron perfiladas por el Dr. E.F.Codd en su escrito de 1972, "Further Normalization of the Data Base Relational Model" (Referente a la normalización de las Bases de Datos Relacionales). La otras regla han sido teorizadas por posteriores matemáticos/Algebristas.

2.4.3 Requisitos del Sistema

Requisitos de recursos de hardware

Las siguientes recomendaciones para el hardware se aplican a matrices de servidores de datos de gran volumen.

- Se recomiendan ocho CPU por cada servidor, como mínimo.
- Se recomienda encarecidamente el uso de redes de área de sistema (SAN, System Area Networks) entre los servidores, aunque no son imprescindibles.

Requisitos de recursos de software

La escalabilidad de la arquitectura de Internet Data Center requiere los siguientes elementos:

- Microsoft Windows® 2000 Advanced Server o Windows 2000 Datacenter Server (preferido).
-

- SQL Server 2000 Enterprise Edition.

Requisitos del sistema

El servidor y los componentes deben estar certificados para usarse con el sistema operativo Windows 2000. Cuando un fabricante obtiene la certificación para un servidor, éste se agrega a la Lista de compatibilidad de hardware (HCL) de Windows 2000.

Antes de instalar la matriz de servidores de datos, debe comprobar que existe una conectividad completa entre los servidores de base de datos y las redes virtuales de área local (VLAN, Virtual Local Area Network).

2.5 Página dinámica.

“ Las páginas dinámicas son páginas HTML generadas a partir de lenguajes de programación (scripts) que son ejecutados en el propio servidor web”(7). A diferencia de otros scripts, como el JavaScript, que se ejecutan en el propio navegador del usuario, los 'Server Side' scripts generan un código HTML desde el propio servidor web.

Este código HTML puede ser modificado -por ejemplo- en función de una petición realizada por el usuario en una Base de Datos. Dependiendo de los resultados de la consulta en la Base de Datos, se generará un código HTML u otro, mostrando diferentes contenidos.

Una página es dinámica cuando realiza efectos especiales o implementa alguna funcionalidad o interactividad. Además, hemos visto que para programar una página dinámica necesitaremos otros lenguajes aparte del HTML. Sin embargo, nunca hay que olvidarse del HTML, ya que éste es la base del desarrollo web: generalmente al escribir una página dinámica el código de los otros lenguajes de programación se incluye embebido dentro del mismo código HTML.

La clasificación de las páginas dinámicas se realiza en función de dónde se lleva a cabo el procesamiento de la página, es decir, el computador que cargará con el peso adicional que supone que la página realice efectos y funcionalidades.

El primer tipo de páginas dinámicas son las del lado del cliente, es decir, las que se ejecutan en el navegador del usuario. Las páginas dinámicas de cliente se escriben en dos lenguajes de programación principalmente: Javascript y Visual Basic Script (VBScript).

Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja, ya que cada navegador tiene sus propias características, incluso cada versión, y lo que puede funcionar en un navegador puede no funcionar en otro.

Como ventaja se puede decir que estas páginas descargan al servidor algunos trabajos, ofrecen respuestas inmediatas a las acciones del usuario y permiten la utilización de algunos recursos de la máquina local.

Podemos hablar también de páginas dinámicas del servidor, que son reconocidas, interpretadas y ejecutadas por el propio servidor. Con ellas se puede hacer todo tipo de aplicaciones web. Desde agendas a foros, sistemas de documentación, estadísticas, juegos, chats, etc. Son especialmente útiles en trabajos que se tiene que acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad los cálculos no se pueden realizar en el ordenador del usuario.

Es importante destacar que las páginas dinámicas de servidor son necesarias porque para hacer la mayoría de las aplicaciones web se debe tener acceso a muchos recursos externos al ordenador del cliente, principalmente bases de datos alojadas en servidores de Internet. Un caso claro es un banco: no tiene ningún sentido que el cliente tenga acceso a toda la base de datos, sólo a la información que le concierne.

Las páginas dinámicas del servidor se suelen escribir en el mismo archivo HTML, mezclado con el código HTML, al igual que ocurría en las páginas del cliente. Cuando una página es solicitada por parte de un cliente, el servidor ejecuta los scripts y se genera una página resultado, que solamente contiene código HTML. Este resultado final es el que se envía al cliente y puede ser interpretado sin lugar a errores ni incompatibilidades, puesto que sólo contiene HTML

Luego es el servidor el que maneja toda la información de las bases de

datos y cualquier otro recurso, como imágenes o servidores de correo y luego envía al cliente una página web con los resultados de todas las operaciones.

Las ventajas de este tipo de programación son que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable.

Como desventajas se puede señalar que será necesario un servidor más potente y con más capacidades que el necesario para las páginas de cliente. Además, estos servidores podrán soportar menos usuarios concurrentes, porque se requerirá más tiempo de procesamiento para cada uno.

*DISPONIBILIDAD
PARA
INTERNET*

3.- Disponibilidad para internet.

3.1. Servidores

Cuando se comparte la conexión a internet o se usa un router de ADSL en modo multipuesto creamos una red interna que está "oculta" al exterior. Mi conexión a internet puede tener IP 62.42.172.3, por ejemplo, y los ordenadores de dentro de la red interna tendrán IP's diferentes, por ejemplo, 192.168.0.1 y 192.168.0.2. Elegir esos números no es casualidad. Las direcciones 192.168.xxx.xxx se usan para redes internas porque esas direcciones IP no se usan para internet, de modo que no hay riesgo de que una IP interna coincida con una IP de internet. Si yo pido una página web desde un ordenador de la red interna, el router o el proxy toman "nota" de quién hace el "pedido" de esa página y cuando llegue se la devuelve al ordenador que la pidió. Es como una aduana.

Cuando tenemos un "servidor", del tipo que sea, dentro de la red interna tenemos un problema: un ftp usa el puerto 21, por ejemplo. Alguien conoce mi IP de internet y hace una petición al puerto 21. La petición llega al proxy/router que es el que controla la IP de internet (o sea, la conexión) pero... ¿a qué ordenador le entrega el router/proxy la petición? Esa petición llega de fuera, no de dentro, luego como no sabe a quién dársela, esa petición "muere en la aduana".

Ahora bien, si hacemos un "mapeado de puertos" la cosa se soluciona. Un mapeado consiste en decirle al "aduanero" (proxy/router) "Mira, colega, cuando alguien te pida conexión al puerto 21, se la mandas a la IP interna 192.168.0.1".

Los servicios de internet más usados usan los siguientes puertos:

- 21 Servidores FTP
 - 22 SSH
 - 23 Telnet
 - 25 Servidores de Correo SMTP
 - 43 Servidores de Correo IMAP4
 - 53 Servidores de Dominio
 - 68 DHCP
-

79	Finger
80	Servidor de Páginas Web
81	Servidor de Páginas Web Seguras
110	Servidores de Correo POP3
137	NetBIOS
139	Servicio de NetBIOS para compartir ficheros e impresoras
161	SNMP
445	Compartir ficheros e impresoras en Windows 2000

Estos puertos se usan "de común acuerdo". Se puede poner un servidor FTP en el puerto 80 y un servidor de páginas web en el puerto 21, pero en ese caso un cliente FTP *NO* podría acceder a nuestro servidor A NO SER que manualmente le indiquemos al cliente FTP que use el puerto 80, ya que por defecto siempre usarán el 21. Del mismo modo, un navegador de internet no cargaría una página web de un servidor que usara el puerto 21 para servir las páginas porque todos los navegadores por defecto piden las páginas al puerto 80.

3.2. IIS Internet Information Services

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un computador en un servidor de internet o Intranet es decir que la computadora que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web). Si se quiere para usar local es mas recomendable utilizar el PWS (Personal Web Service) El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Existen otros servidores web que pueden usarse como alternativa a este que es de propiedad de Microsoft Corporation, por ejemplo Apache, Cherokee que son desarrollados en Software Libre y otros muchos.

IIS es la versión más reciente de Server Web para NT4, gestiona aplicaciones FTP y WEB. Entre sus características destaca la rica dotación de instrumentos satélite al servidor WEB, así como características de arquitectura completamente renovadas respecto a las versiones precedentes.

Una de las características más importantes es la presencia del protocolo HTTP 1.1(Hypertext Transfer Protocol) que ofrece sensibles mejoras de las prestaciones, disminuyendo los tiempos de respuesta en la transmisión. Los nuevos Defaults del protocolo HTTP1.1 residen en algunos de los elementos que lo componen, tales como el Pipeling, las conexiones persistentes, las transferencias por bloques CHUNKED, el soporte para el proxy.

3.2.1 Hardware

Los requisitos de hardware, para un funcionamiento óptimo de Internet Information Server, dependen de las exigencias específicas del servidor, del número potencial de usuarios conectados, de eventuales interconexiones con bases de datos, del uso de ASP, LOG.. Es asimismo conveniente tener en cuenta posibles aplicativos que pudieran reducir la velocidad de actuación del servidor IIS. Es aconsejable, por tanto, evitar la instalación en PDC (Primary Domain Controller), que estén ya abrumados de trabajo debido a la convalidación de los usuarios. Si es posible, debería configurarse un Member Server dedicado exclusivamente al uso de IIS. Esto vale también para servidores que albergan aplicativos como SQL Server.

Dimensionando pues una tipología de hardware específica para un cierto número de Accesos / Clientes.

□ **200 Accesos / Clientes**

Intel Pentium 100MHz

De un mínimo de 32 MB a 64 MB RAM

60 MB de espacio en disco duro para la instalación

De un mínimo de 250MB a 2GB de espacio libre en el disco duro para el Caching

□ **de 200 a 2000 Accesos / Clientes**

Intel Pentium 133MHz

64 MB RAM mínimo

60 MB de espacio en disco duro para la instalación

De un mínimo de 2GB a 4GB de espacio libre en el disco duro para el Caching

□ **más de 2000 Accesos / Clientes**

Intel Pentium 166MHz mínimo

64 MB RAM mínimo

60 MB de espacio en disco duro para la instalación

De un mínimo de 2GB a 6GB de espacio libre en el disco duro para el Caching

DEC ALPHA 150MHz mínimo

64 MB RAM mínimo

60 MB de espacio en disco duro para la instalación

De un mínimo de 300MB a 2GB de espacio libre en el disco duro para el Caching.

La instalación de varios "sitios" en un único servidor, comporta una empeoramiento proporcional de las prestaciones globales. Es aconsejable, por tanto, apoyarse, sobre todo en el caso de INTRANET, en más de una dirección IP alojándolas en varias tarjetas (Multihomed), antes que redimensionar las bandas de red según las solicitudes potenciales de cada sitio (Bandwith Throttling).

3.3 Apache

“ El **servidor HTTP Apache** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor *parcheado*)”.(8)

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

En la actualidad (2005), Apache es el servidor HTTP más usado, siendo el servidor HTTP del 68% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft.)

Versión 2.x

El núcleo 2.x de Apache tiene varias mejoras clave sobre el núcleo de Apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPv6. Algunos ejemplos son:

- [Jakarta Tomcat](#)
 - [Jakarta Struts](#)
-

3.4 IIS VS APACHE

Los dos servidores de la red más difundidos son, sin duda alguna, IIS de Microsoft y el adversario Apache, nacido de las cenizas de un anterior proyecto y de un grupo de voluntarios.

La diferencia principal entre los dos apliques es la plataforma para las que fueron pensadas: IIS para servidores que se basan en los sistemas Windows, Apache para la gran familia de los Unix. Sin embargo, un punto a favor de Apache puede ser el esfuerzo (en parte también logrado) para llevar el servidor de la red a plataformas no nativas, como, por ejemplo, los sistemas Windows. Obviamente, los mejores resultados se obtienen en plataformas nativas, aunque, se murmura, Apache no se porta mal en otras plataformas.

La "lucha" entre estos dos servidores de la red siempre ha sido abierta, y a ésta siguió también un fuerte dualismo entre los dos sistemas operativos que los respaldan: Windows, por una parte, los Unix por la otra. Y, como todos los dualismos, cada uno afirma, datos a la mano, que su producto es mejor que el del competidor.

La rivalidad de estos dos sistemas operativos continuará por muchos años. Ambos servidores tienen sus propuestas, pero esto realmente depende del propósito del servidor web. Las diferencias relevantes se muestran en la tabla 3.1.

Tabla 3.1 Comparativa entre IIS y Apache

Apache	IIS
Pro's <ul style="list-style-type: none"> • Estable • Muy rápido • libre 	Pro's <ul style="list-style-type: none"> • Uso amigable • Fácil de programar con ayuda de ASP • Buen soporte
Contras <ul style="list-style-type: none"> • Difícil de manipular si no hablas you don't speak geek • Requiere ser compilado 	Contras <ul style="list-style-type: none"> • Debes comprar una copia legal • Problemas de seguridad(eg. Code Red)

3.4.1 Fundamentos

IIS ha sido un componente opcional del sistema operativo de Windows Server a partir de Windows NT 4.0. Recordando entonces, éste era un servidor d web básico en la forma de IIS 3.0. Un paquete optativo fue realizado rápidamente después de Windows NT 4.0, incluyendo el IIS 4.0 y fue la primera versión “real” de IIS para ser usada como plataforma de servidor web.

IIS 5.0 fue incluido en Windows 2000 server y versiones de escritorio, y la versión actualizada, IIS 5.1, se incorporó en Windows.xp. Los cambios entre IIS 4 e IIS 5 son muy pocos y de menor importancia.

Windows 2003 server incluye el nuevo componente IIS 6.0, el cuál es una reescritura casi completa de las ediciones anteriores. Éste soporta un

nuevo modelo de ejecución, mejor administración, y un significativo incremento en el crecimiento del mismo.

Apache, por otra parte, tiene una herencia más vieja, basado originalmente en el código del httpd , algunos discuten sobre si éste fue el primero en iniciar la revolución entera del Web. Apache 1.3.x, que era la versión de la producción hasta 2002, fue el producto más usado en su tiempo y en numerosas ocasiones se usó dentro de otros sistemas operativos, incluyendo Windows.

El lanzamiento de la producción de Apache 2.0 golpeó los estantes en 2002. El nuevo lanzamiento incorporó un ambiente a estrenar de la ejecución que separó la funcionalidad de la base del sistema de Apache del sistema que apoyó y procesó peticiones. Apache 2.0 ahora se apoya bajo amplia gama de sistemas operativos potenciales, incluyendo todas las versiones de Windows, Linux, Unix, y Mac OS X.

3.4.2 Ambiente de ejecución

IIS y Apache funcionan muy diferente y tienen así una variedad de ventajas y desventajas. IIS está diseñado y disponible para trabajar solamente dentro del ambiente de Windows. Con IIS 6.0, la única plataforma que lo soporta actualmente es el servidor 2003 de Windows. Aunque esto limita las plataformas del despliegue para los servicios de Web, basados en el IIS provee de numerosos beneficios , incluyendo una buena cooperación con el sistema operativo , mejor control y funcionamiento con la variedad de herramientas y utilidades del mismo.

En IIS 6.0, la cooperación entre el sistema operativo y el servidor es mayor . En contraste a las versiones anteriores, el componente que acepta peticiones de clientes y los procesa ahora se convierte en dos componentes separados. El oyente del modo del núcleo, HTTP.sys, escucha y acepta peticiones de clientes, situando dichas peticiones en una o más colas de espera. IIS entonces procesa las peticiones que están en espera utilizando un trabajo menor para controlar la ejecución de peticiones y aplicaciones individuales.

Este proceso separado permite que las peticiones sean aceptadas incluso cuando los procesos de trabajo de IIS técnicamente no están funcionando, y también permite un control más fino en los procesos del trabajador que manejan peticiones. Así, el admin (o el servidor, automáticamente), puede reciclar y recuperar peticiones sin necesidad de encender el servidor o en tal caso reiniciarlo.

Apache 2.0 era una reescritura importante de las versiones anteriores. Entre los muchos cambios, el servidor del Web está disponible ahora en una variedad de plataformas, incluyendo Windows. El rediseño de Apache permite soportar una gran gama de plataformas de manera más eficiente que los modelos de ejecución de Windows o Unix y así obtiene un mejor rendimiento del sistema operativo.

La base del sistema es el tiempo de pasada portable de Apache (APR), permite a Apache funcionar en más o menos cualquier sistema con un compilador de C. Un número de módulos del multi-processing (MPMs) entonces proporcionan el soporte para atender y procesar las peticiones.

3.4.3 Componentes Dinámicos

El ambiente dinámico primario para el desarrollo dentro de IIS es páginas activas del servidor. Esto es un término genérico para una solución que permita que el código sea encajado en las páginas del HTML. Estas páginas del ASP son analizadas por el servidor antes de ser provisto al cliente como HTML. El sistema del ASP permite que los reveladores trabajen en un número de diversos idiomas, incluyendo VISUAL BASIC, VBScript, Javascript, Java, y C/C++, junto con otras alternativas abiertas de la fuente, tales como Perl y python. Además, IIS continúa apoyando métodos tradicionales del cgi junto con su propia habitación de los sistemas de la filtración y de la ejecución en la forma de filtros de ISAPI.

Apache también es diseñado para trabajar con una amplia gama de idiomas, vía el modelo del cgi, o con el uso de módulos dinámicos

directamente incorporando al intérprete de la lengua en el ambiente de Apache. Esto aceleró perceptiblemente la ejecución de los componentes dinámicos para las idiomas como PHP, el Perl, y el python.

Ambos sistemas apoyan las páginas del servidor de Java modelan, y es posible emigrar la mayoría de los usos de JSP entre las dos plataformas con pocos cambios. Otros idiomas pueden alcanzar esto con resultados que varían. Incluso el ASP se puede apoyar debajo de Unix a través del componente de ChilliSoft ASP , a través del módulo de Apache::ASP, o del módulo de Apache

Un elemento que no se puede emular actualmente bajo Unix es el ambiente de Microsoft.NET. IIS 6.0 y el uso pesado de la marca del servidor 2003 de Windows y proveen de niveles excelentes de la integración el marco del NET. Las características de cada servidor se muestran en la tabla 3.2

Tabla 3.2 Características de servidor de IIS y Apache

Característica	IIS	Apache
ASP	Sí	Con Chilisoft, Apache::ASP, o modmono
Cgi	Sí	Sí
Perl	Sí	Sí
Python	Sí	Sí
PHP	Sí	Sí
JSP	Sí	Sí
NET Integrado	Sí	No

3.4.4 Seguridad y autenticación

IIS se beneficia de la integración cercana con el sistema operativo. El mismo sistema de la configuración del usuario y del grupo que acciona el

sistema principal de la autorización dentro de Windows también proporciona los servicios de la seguridad y de la autenticación para el sistema principal de IIS. Reduce los gastos indirectos de gerencia apoyando solamente un sistema, y permite al sistema admin abrir una sesión automáticamente a usuarios a través de un Intranet, asumiendo que ellos se identificaron ya dentro del ambiente de Windows.

Porque el mismo sistema se utiliza a través del sistema operativo, puede también ser utilizado para definir seguridad y para tener acceso a diversos componentes del sistema de ficheros en el cual el sitio del Web confía. Por ejemplo, cuando un usuario entra al sitio del Web, su capacidad de hojear los directorios dentro del sitio es definida por un sistema de estructuras del usuario y del grupo.

El sistema de la seguridad y de la administración de Apache no está bien integrado con el soporte del sistema operativo. Aunque hay módulos y adaptadores que apoyan una variedad de diversas fuentes de la autenticación y de seguridad, incluyendo directorio activo, el archivo del password de Unix, y los servidores de LDAP, el servidor que resulta se aísla relativamente del sistema que apoya. Por ejemplo, cuando un usuario autentifica contra el archivo del password de Unix, Apache todavía no hace caso de los permisos del archivo en el sistema de ficheros subyacente.

En términos de transacciones seguras, ambos sistemas apoyan tecnología del cifrado en el SSL, y se pueden utilizar con las puestas en práctica de IPsec e IPv6. Las características de seguridad se muestran en la tabla 3.3.

Tabla 3.3 Características de seguridad de IIS y Apache

Característica	IIS	Apache
Asegure La Conexión	Sí	Sí
SSL	Sí	Sí
Autenticación Básica	Sí	Sí
Autenticación Del Resumen	Sí	Sí
Autenticación de LDAP	Sí	Sí
Autenticación Activa Del Directorio	Sí	Solamente con los módulos de tercera persona
Autenticación Del Pasaporte	Sí	No

3.4.5 Confiabilidad

Esta última comparación es probablemente la más discutible al comparar IIS y Apache. De acuerdo con la experiencia de este escritor, hay diferenciación muy pequeña entre las versiones más recientes. En el laboratorio, una caja del servidor 2003 de Windows ha estado funcionando en un sitio del Web desde enero de 2003 (cuando el sistema fue instalado) - casi ocho meses de sólido en el tiempo de la prensa. Todas las actualizaciones necesarias se han aplicado sin el recomienzo del servidor. Por la comparación, la máquina de Apache/Solaris 8 al lado de ella ha estado funcionando por 335 días sin la edición.

IIS 6.0 ahora proporciona sistemas extensos de la gerencia y de control sobre los procesos del trabajador que controlan la mayoría de las peticiones. Para mejorar la confiabilidad, IIS puede también dividir en compartimientos los usos, dándoles su propio espacio de la memoria y de la ejecución, o usando un espacio compartido con otros usos. Esto significa que cuando un uso causa un problema, muere dentro del proceso del trabajador y no afecta cualquiera de los otros usos o de la base de proceso de la petición en el proceso.

Apache maneja automáticamente mucho de los aspectos de la gerencia del proceso y del hilo de rosca, pero el aislamiento de la memoria y del uso sigue siendo una edición que no se ha resuelto ni se ha cubierto con eficacia. Apache evita algunos de los problemas automáticamente reciclando componentes, con la base continuamente funcionando incluso en el acontecimiento de una falta seria en uno de los componentes. Hay situaciones inmóviles, sin embargo, donde Apache puede necesitar ser recommenzado.

Un ejemplo de esto es la configuración del sistema de Apache a sí mismo. Para un sitio del Web que funciona de un servidor IIS-basado, la mayoría de las actualizaciones y de las mejoras no tendrán ningún efecto en la capacidad de los usuarios de tenerles acceso porque se realizan los cambios de configuración mientras que el sistema todavía está funcionando. Apache, sin embargo, requiere un reinicio teórico.

Eso dicha, el uso de htaccess dentro de Apache significa que muchos artículos que necesiten normalmente la modificación con el curso de la vida de un sitio del Web se puede manejar dinámicamente.

Característica	IIS	Apache
Gerencia De Process/Thread	Sí	Sí
Usos Aislados	Sí	Limitado
El Corregir Vivo De la Configuración	Sí	No

A pesar del arsenal de diferencias entre los dos sistemas, el elegir entre ellos viene bajo a las necesidades y a los requisitos de la organización y, en un grado inferior, de las necesidades y de las preferencias personales de éstos que la usan. Por ejemplo, muchos reveladores prefieren trabajar en el Perl en un ambiente Unix-basado de Apache, aunque el Perl y Apache están disponibles bajo Windows, y el Perl se apoya por separado debajo de IIS.

Si la gerencia del servidor es una preocupación primera, entonces las capacidades extensas de la gerencia de IIS pueden ser la manera a utilizar. El aspecto de la seguridad y la integración con el directorio activo, y por lo tanto el resto de los mecanismos de la seguridad, es también una ventaja importante.

Sin embargo, es importante hacer caso del costo de la edición del despliegue. Apache está libre, y se puede instalar en OS's como Linux que esté también libre. IIS está disponible solamente como parte del servidor 2003 de Windows. Aunque un servidor nuevo 2003, edición de Windows del Web está disponible en un descuento significativo comparado a su primo estándar de la edición, todavía está, en lo menos en la superficie, más costosa que Apache.

*DISEÑO DE
EXAMENES EN LINEA
PARA LA CARRERA DE
INGENIERIA MECANICA*

4.- Diseño de exámenes en línea para la carrera de Ingeniería Mecánica.

4.1 ¿ Cómo generar la base de datos?

Para nuestro uso iremos desarrollando paso a paso la base de datos que utilizamos y la llamaremos “ingepimer”. Teniendo como sistema operativo Windows 2003 server hemos instalado también SQL Server 2000.

Hay dos métodos para realizar la base de datos. La primera es de forma gráfica y la segunda por medio de comandos de SQL que es una herramienta muy poderosa y que nos va a facilitar la programación de las páginas web que más adelante explicaremos.

Comencemos entonces a generar la base de datos primero de manera gráfica.

Antes que nada iniciamos SQL Server en administrador corporativo desglosando el árbol hasta donde aparece el archivo: bases de datos. Damos click derecho y escogemos la opción de “Nueva base de datos”, asignándole el nombre en el cuadro de texto inicial: ingepimer.

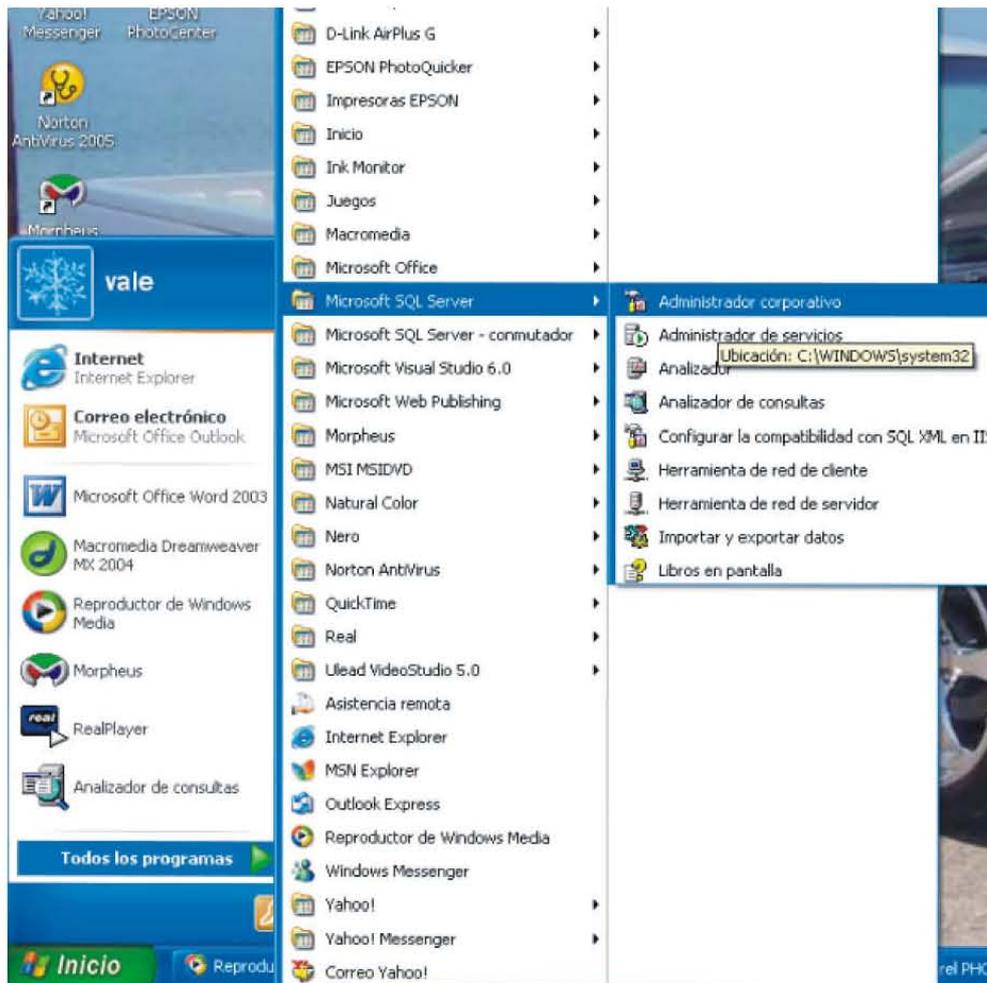


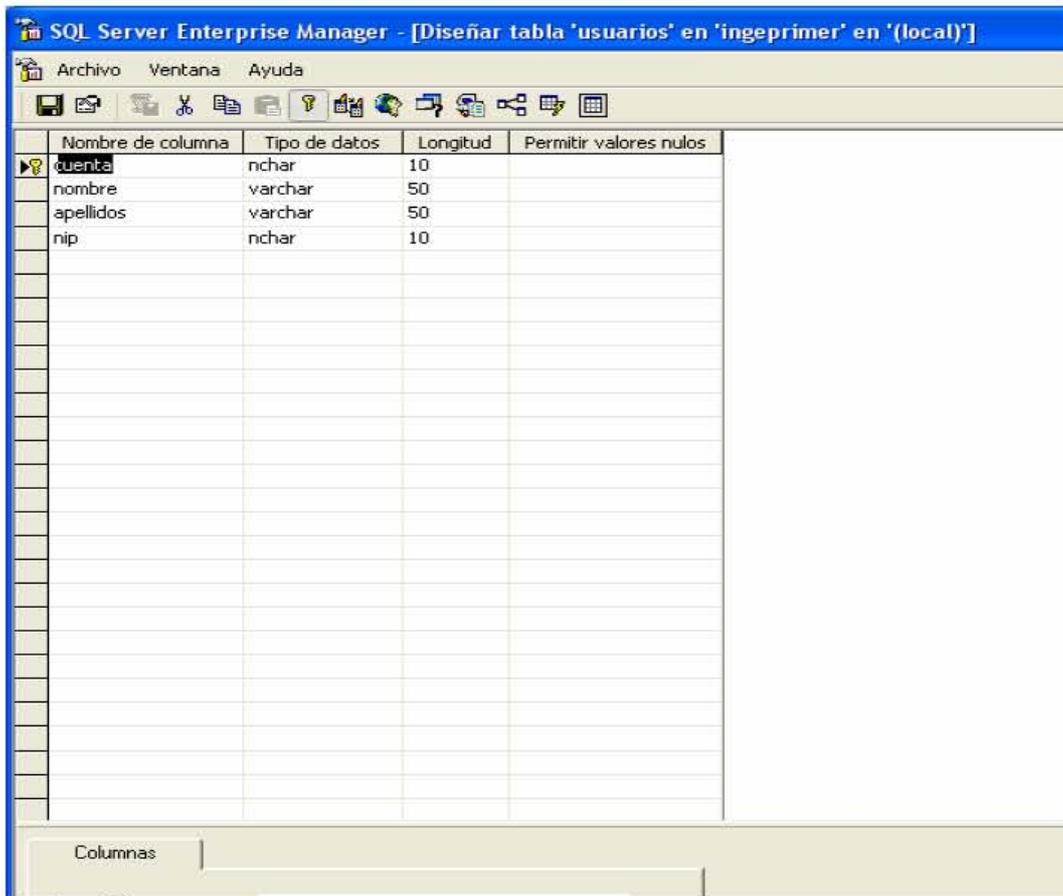
Fig 4.0 Cómo entrar al ambiente SQL



Fig 4.1 Cómo generar la base de datos

Ahora nos posicionamos en la base de datos que acabamos de crear y nos dirigimos al ícono “tablas” para que de manera gráfica generemos éstas, en las cuales guardaremos toda la información de nuestro sistema.

Dentro de Ingeprimer volvemos a dar click derecho y ahora seleccionamos la opción nueva tabla y entonces comenzamos a generar los campos de la tabla recordando establecer la llave primaria a las tablas que lo necesiten. De este modo generamos todas las tablas que sean necesarias para la base de datos “Ingeprimer” de nuestro sistema.



Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
cuenta	nchar	10	
nombre	varchar	50	
apellidos	varchar	50	
nip	nchar	10	

Fig 4.2 Una tabla en la base de datos ingeprimer

Después de haber construido y normalizado todas las tablas de nuestra base ingeprimer tenemos que ligarlas de acuerdo a la forma en que cada una se deba utilizar, ya sea uno a uno, uno a varios , varios a varios.

El diagrama de nuestra base de datos ya terminada es el siguiente

La programación de la base de datos se hace desde el analizador de consultas a partir de comandos de SQL con los cuales obtenemos el mismo resultado y es una forma más eficiente de realizar el trabajo.

Creamos la base de datos.

```
CREATE DATABASE ingeprimer
```

Con este ejemplo creamos las tablas de la base de datos.

```
CREATE TABLE usuarios  
(  
  Cuenta nchar (10)primary key,  
  Nombre varchar(50),  
  Apellidos varchar (50),  
  Nip nchar (10)  
)
```

Se puede notar que dentro de cada campo lleva las especificaciones del mismo de acuerdo a las necesidades para las que se van a utilizar dichos campos.

4.1.1 Consultas a la base de datos

Las consultas hacia la base de datos es importante pues por medio de éstas puedes visualizar el contenido de tu base, puedes añadir datos así como también eliminar datos y realizar consultas específicas si en tu base de datos no existe redundancia.

Par visualizar el contenido de una tabla de la base de datos utilizamos el comando SELECT, con el cual podemos realizar consultas más específicas.

```
SELECT * FROM usuarios
```

```
SELECT cuenta FROM usuarios WHERE nombre = 'Luis Alberto'
```

A través del comando INSERT iremos guardando los datos que queramos en la base , por ejemplo en la tabla usuarios que es la anterior insertaremos el nombre: Luis Alberto Reyes Juárez con cuenta 96342614 y nip 08031979.

```
INSERT INTO usuarios  
VALUES (96342614,Luis Alberto, Reyes Juárez,08031979)
```

Entonces la tabla usuarios ya contiene un elemento en ella.

El comando *DELETE* sirve para borrar datos de las tablas que contiene la base de datos.

Para borrar una columna específica

```
DELETE FROM usuarios WHERE cuenta = '96342613'
```

Para borrar todos las columnas de la tabla

```
DELETE FROM usuarios  
or  
DELETE * FROM usuarios
```

El comando UPDATE modifica algún dato de la columna dentro de una tabla .

```
UPDATE usuarios  
SET cuenta = 98362412  
WHERE nombre = 'Luis Alberto'
```

El comando JOIN sirve para hacer referencia a dos tablas de la base de datos donde se tienen columnas con datos iguales.

```
SELECT usuarios.cuenta, usuarios.nombre, usuarios.apellidos,  
calificaciones.calificacion1, calificaciones.calificacion2,
```

calificaciones.calificacion3, calificaciones.examenfinal FROM usuarios,calificaciones WHERE usuarios.cuenta=calificaciones.cuenta

El ejemplo anterior combina las tablas usuarios y calificaciones para mostrar gráficamente las calificaciones de cada uno de los usuarios que realizaron el examen en línea. Esta consulta se utiliza en la rama del profesor, que podrá ver a los alumnos con sus respectivas calificaciones del parcial correspondiente.

A continuación mostramos gráficamente el uso de este comando utilizando el analizador de consultas del SQL Server 2000.

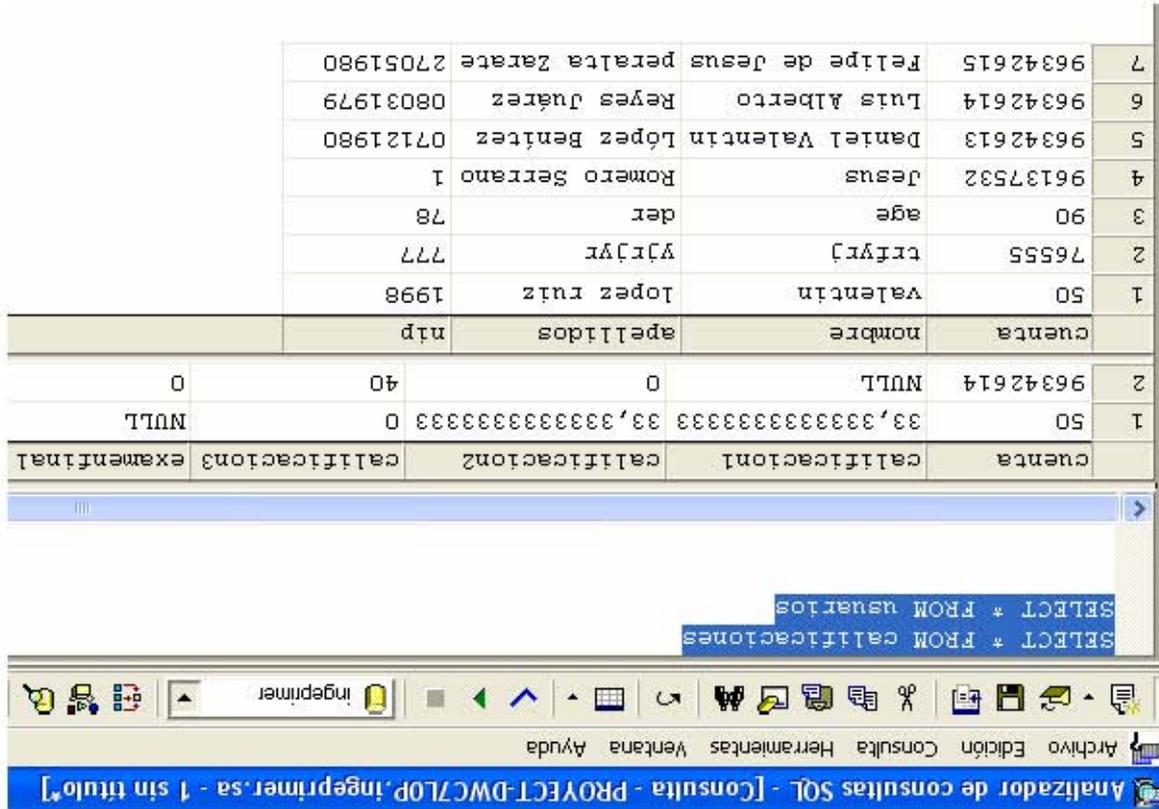


Fig 4.4 Consulta de información

El resultado de la consulta anterior se muestra en la siguiente figura, donde podemos ver claramente cómo de las dos tablas anteriores, obtenemos sólo una con los datos que necesitamos.

Analizador de consultas SQL - [Consulta - PROYECT-DWC7/LOP.ingeprimer.sa - 1 sin título*]

Archivo Edición Consulta Herramientas Ventana Ayuda

ingepimer

```

SELECT usuarios.cuenta, usuarios.nombre, usuarios.apellidos, calificaciones.calificacion1,
calificaciones.calificacion2, calificaciones.calificacion3, calificaciones.examenfinal
FROM usuarios,calificaciones
WHERE usuarios.cuenta=calificaciones.cuenta

```

	cuenta	nombre	apellidos	calificacion1	calificacion2	calificacion3	examenfinal
1	50	valentin	lopez ruiz	33,33333333333333	33,33333333333333	0	NULL
2	96342614	Luis Alberto	Reyes Juárez	NULL	0	40	0

Fig 4.5 Consulta específica

4.2. Conexión de páginas ASP con una base de datos

Para lograr una interactividad de nuestras páginas ASP es necesario integrar la base de datos que hemos creado, con la finalidad de que los usuarios puedan consultar desde la web dicha base y realizar todas las tareas requeridas.

La forma de establecer la conexión con ASP utilizando SQL Server 2000 y sobre un sistema operativo Windows que trabaja con IIS se muestra a continuación.

```
<% @LANGUAGE="VBSCRIPT" %>
```

```
<%
```

set cnn=server.CreateObject("ADODB.Connection") En esta línea se especifica el tipo de conexión que vamos a realizar: "ADODB.Connection"

ConStr="provider=SQLOLEDB.1;datasource=PROYECT-dwc7l0p;USER ID =sa; password=sa;initial catlog=ingepimer" ésta es la cadena de connexion donde especificamos todos los datos del servidor(proveedor,nombre de la máquina, identificador, contraseña, nombre de la base de datos).

cnn.open ConStr,"sa","sa" Mandamos a abrir la cadena de conexión

cnn.defaultdatabase="ingepimer"

*set rs=cnn.execute("select * from calificaciones order by cuenta")* Una vez establecida la conexión podemos realizar un sinfín de consultas a nuestra base de datos desde la Web

%>

4.3 Diseño de página dinámica

De acuerdo al objetivo planteado en éste proyecto el cual es desarrollar un sistema de exámenes en línea, establecimos en base a las necesidades del diseño tres tipos de usuario: 1. profesor, 2. alumno, 3. administrador. Cada uno de estos usuarios tendrá un acceso independiente pero el alumno no puede ver información detallada del profesor o del administrador, mientras que el profesor puede ver los datos de identificación del alumno y calificaciones, y el administrador tendrá ciertos privilegios en la base de datos, como agregar y eliminar usuarios, modificar exámenes etc.

Todos los usuarios interactúan con la base de datos "ingepimer" ya que se necesitan validaciones de usuario, modificar datos y obtener resultados de la misma. Esta acción convierte a nuestro sistema en un sistema dinámico.

En el diseño de cada ventana de usuario se buscó la mayor seriedad posible en fondos e imágenes, pues el alumno al realizar el examen no debe tener ningún tipo de distracción y que cada componente sea lo más específico posible.

A continuación presentamos un diagrama en el cual se visualizan las acciones y restricciones de cada usuario. Este diagrama es la base de nuestro sistema.

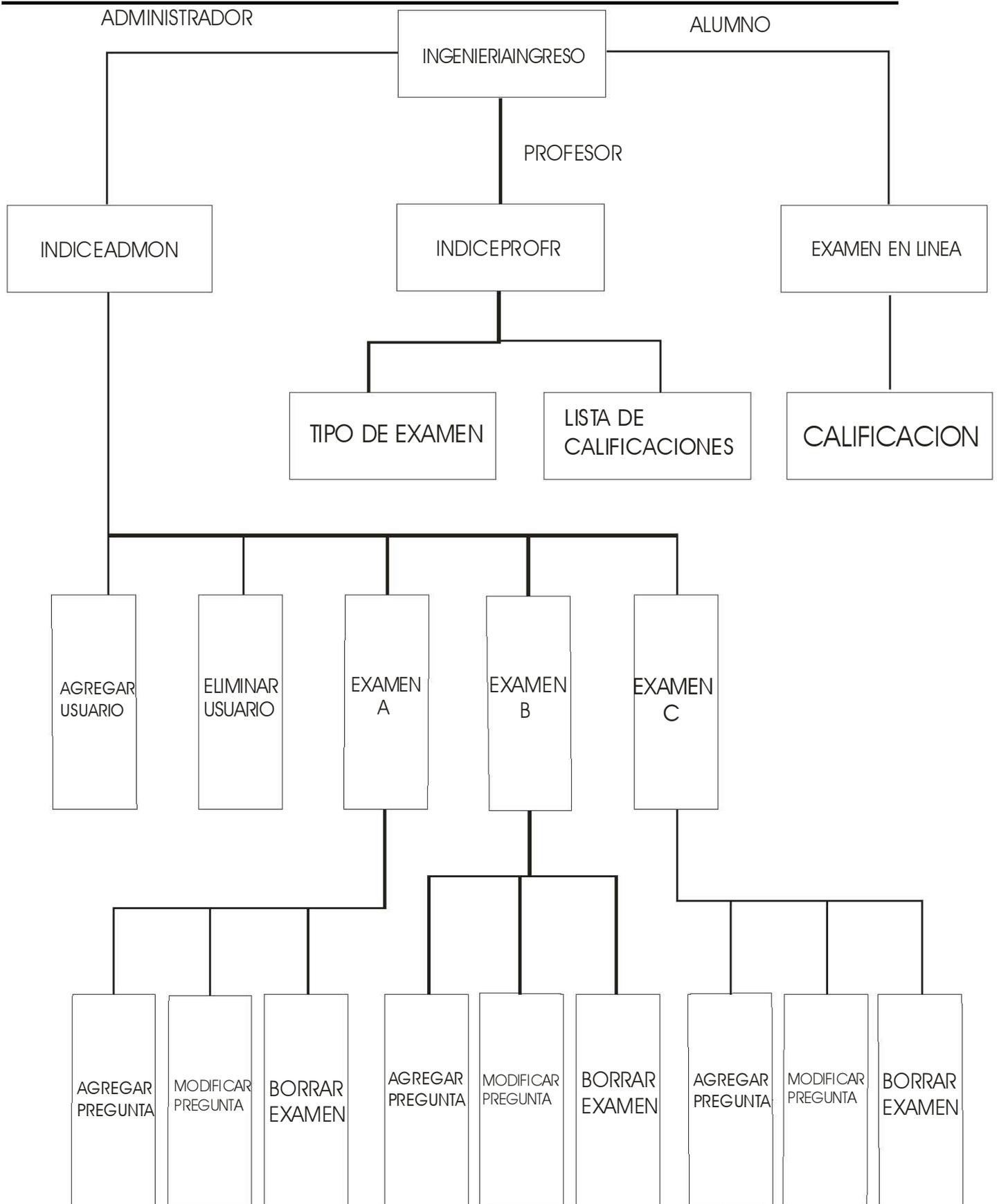


Fig 4.6 Diagrama del sistema

4.4. Implementación.

En este subtema explicaremos de manera específica cómo se fueron realizando cada una de las páginas dinámicas de cada usuario y la página de inicio.

Los lenguajes utilizados para la implementación son: HTML para la elaboración de las páginas Web, SQL Server 2000 para la elaboración de la base de datos del sistema, Visual Basic Script y ASP para lograr comunicación con la base de datos, con lo cual se vuelve un sistema dinámico.

INGENIERIAINGRESO.ASP

Ésta es la página de inicio de nuestro sistema en la cual para ingresar se pide un login y un password, que en nuestro caso es el número de cuenta del alumno y un NIP que el administrador otorga respectivamente. Con estos datos, se hace una consulta a la base de datos en la cual se verifica si el usuario está registrado en ella. Si el usuario sí está registrado, tendrá acceso al sistema y con los datos ingresados la página dinámica decidirá a que página de usuario corresponde. Si no está registrado, se manda un mensaje de acceso denegado.

Todas estas consultas y toma de decisiones no las realiza INGENIERIAINGRESO:ASP, ésta sólo captura datos que son enviados a la página siguiente llamada “validalogin” que es la que realmente realiza todo el proceso de validación y nadie puede ver.

El código de la página de inicio lo presentamos a continuación con la finalidad de que sirva de ejemplo para posteriores actualizaciones.

<HTML >

<HEAD >

<title>Ingeniería.unam</title>

</HEAD>

<BODY background="C:\Documents and Settings\vale\Mis documentos\Mis imágenes\correos\logo2.bmp" text="#0000CC" >

```
<DIV align=center>
```

```
<P><FONT face=Forte size=+4 color="#0000FF">Bienvenido al
sistema de
```

```
exámenes en línea de Ingeniería</FONT> </P>
```

```
<P align="center">
```

```
<form action="validalogin.asp" method="post" >
```

```
<P>
```

```
; <font size="+1">LOGIN</font>
```

```
<input type="text" name="cuenta" style="WIDTH: 62px; HEIGHT:
22px" size=7 maxlength="8" ><BR>
```

```
; <font size="+1">NIP</font> <input type="password"
name="NIP" style="WIDTH: 62px; HEIGHT: 22px" size=9
maxlength="8">
```

```
<BR>
```

```
<br>; <INPUT type="submit" value="enviar" > <br></P>
```

```
</form></DIV>
```

```
</BODY>
```

```
</HTML>
```

También mostramos un ejemplo de cómo se validan los ingresados.

```
<% dim usuario, password
```

```

usuario=request.form("usuario")

password=request.form("password")

set cnn= Server.createobject("ADODB.connection")

constr="provider=SQLOLEDB.1;datasource=sistema05; user ID=fx;
password=fx;

initial catlog=usuarios login"

cnn.openconstr,"fx"fx"

cnn.default database="usuarios login"

response.write("conexión conseguida")

setrs=cnn.execute("select usuario, password from login where
usuario=" &usuario&"and password=" &password"")

if (rs.eof) then

    response.write("<h1>" &"acceso denegado" &"</H1>")

else

    response.write"bienvenido" &rs("usuario")

end if

set rs nothing %>

```

INDICEADMON

Ésta página es la primera página que visualiza el usuario "administrador" después de ingresar sus datos. Las aplicaciones que el administrador tiene son: agregar usuario, eliminar usuario, modificar o crear tipos de examen, especificar en qué período de exámenes nos encontramos. Esto quiere decir que el administrador lleva un completo control del sistema ya que él puede acceder de forma directa a la información contenida en la base de datos.

En ésta página se presenta la primera consulta a la base de datos “ingepriemer” pues se visualiza la tabla con los usuarios que están registrados. Esta visualización se realiza de la siguiente manera.

```
<% set rs=cnn.execute("select *from usuarios order by cuenta ")

    if not rs.EOF then

        do while not rs.EOF

            response.Write("<tr><td>"&rs("cuenta")&"</td>"&"<td>"&rs("n
ombre")&"</td>"&"<td>"&rs("apellidos")&"</td>"&"<td>"&rs2("clav
e")&"</td></tr>")

            rs.movenext

        loop

    end if %>
```

INDICEPROFR

Indiceprofr.asp es la primera página que ve el usuario profesor después de haber ingresado sus datos. La elaboración de ésta página es similar a la de indiceadmon, con la diferencia en las tareas que puede realizar. El profesor sólo puede consultar los tipos de examen y las calificaciones con la finalidad de que el profesor no pueda modificar valores en la base de datos.

EXAMENLINEA

Ésta página es la primera y la única que visita el usuario “alumno” con la posibilidad de interactuar con la base al momento de resolver el examen. Cuando termina su examen envía las respuestas a una tabla que es comparada con las respuestas correctas que contiene la base de datos. Después de esto el sistema envía el resultado de la comparación que es la

calificación obtenida del examen, mientras el sistema guarda la calificación en la base de datos “ingepriemer”.

Hay que mencionar que este código se utilizó en todos los tipos de examen puesto que adoptan la misma ideología .El algoritmo con el que se hace la comparación de las respuestas se da a continuación.

```
<% dim mitabla  
  
dim I,J  
  
mitabla= rs.GetRows  
  
dim denominador  
  
dim a  
  
dim iA  
  
dim g  
  
dim h  
  
dim k  
  
dim calificacion  
  
a=0  
  
k=0  
  
iA=1  
  
for I = 0 to UBound(mitabla,2)  
  
'Abrimos una nueva fila  
  
'Response.Write("<TR>")  
  
'Desde el primero hasta el último "campo"...  
  
for J = 0 to Ubound(mitabla, 1)
```

```
'Imprimo una celda para cada campo
'Response.Write("<TD>" & mitabla(J, I) & "</TD>")
g=CInt(mitabla(J,I))
h=CInt(request.Form("&iA&"))
'response.Write(h)&"<br>"
if h=g then
k=k+1
    'response.Write("1")
else
    a=a+1
end if
iA=iA+1
next
'Response.Write("</TR>")
next
'Response.Write("</TABLE>")

denominador=k+a
calificacion=(k/denominador)*100 %>
```

AGREGARUSUARIO

La página “agregarusuario” es una liga de la página del administrador en donde podemos registrar distintos usuarios, introduciendo sus datos de

acuerdo al perfil que queramos darle a cada usuario. Los datos que introducimos son: número de cuenta, nombre y apellidos, número de identificación personal, tipo de usuario, tipo de examen (si el usuario lo requiere). Estos datos son enviados a la base de datos, los cuales de acuerdo a las especificaciones hechas, ayudarán para diferenciar al tipo de usuario que ingresa al sistema.

Un ejemplo de cómo interactúa la base de datos con la página es:

```
<html>

  <head>

    <title> agregar</title>

  </head>

  <body>

    <form action="procesa.asp" method="post">

      Clave: <input type="text" name="clave"><br>

      Nombre: <input type="text" name="nombre"><br>

      Apellidos: <input type="text" name="apellidos"><br>

      <input type="submit" value="enviar"><br>

    </form>

  </body>

</html>
```

Y el proceso de inserción de datos es:

```
<%set cnn= Server.createobject("ADODB.connection")
```

```
constr="provider=SQLOLEDB.1;datasource=sistema05; user ID=fx;
password=fx;
```

```
initial catalog=usuarios login"
```

```
cnn.openconstr,"fx""fx"
```

```
cnn.default database="usuarios login"
```

```
set rs= cnn.execute("insert into
usuarios(clave,nombre,apellidos)values('"&request.form("clave")&"',
"&request.form("nombre")&"', "&request.form("apellidos")&"')")
```

```
cnn. Close
```

```
response.redirect("default.asp")%>
```

ELIMINARUSUARIO.ASP

En ésta página se habilita la tarea de eliminar a un usuario de la base de datos cualquiera que éste sea, y se realiza a través de introducir el número de cuenta del usuario y enviarlo a la base de datos para que se localice el dato y se borre. Eliminar un usuario es útil pues muchas veces el alumno se da de baja temporal, y si no se eliminara podría tener acceso al examen sin que estuviera inscrito en la asignatura.

El código para eliminar un usuario es prácticamente el mismo pero con la diferencia en la consulta a la base de datos pues en lugar de insertar un dato estamos borrando uno.

EXAMEN

En éste apartado tratamos a tres páginas al mismo tiempo pues sus consultas repercuten a los tipos de exámenes en si. Las tareas que podemos hacer con éstas tres páginas son comunes y se describen a continuación.

Agregar pregunta.- en ésta página es donde introducimos las preguntas que queremos que el usuario “alumno” visualice. Debemos introducir el número de pregunta, tipo de examen, pregunta, respuestas opcionales donde una de ellas es la respuesta correcta, y respuesta correcta con el fin de realizar la comparación de las respuestas que el alumno dará con la respuesta correcta almacenada en la base de datos como lo explicamos anteriormente. Enviamos todos estos datos a la base “ingepimer” para que queden almacenados.

Modificar pregunta.-la página de modificar pregunta se instaló pues creímos si el examen sólo variara en una o dos preguntas, se pudieran modificar éstas sin necesidad de borrar todo el examen y tener que reescribirlo otra vez.

Borrar examen.- borrar examen se utiliza para realizar un cambio completo del examen en la base de datos. Ésta consulta borra todas las preguntas y respuestas en nuestra base de datos del examen del tipo en que estemos consultando.

4.5. Resultados del diseño

Los resultados de la implementación son notables, pues se logra una interfaz amigable y efectiva que hasta el momento no ha tenido errores. Tratamos de abarcar las modificaciones y consultas necesarias a nuestra base de datos, logrando que los operarios no truncaran la comunicación entre Web y base de datos al cometer un error en la captura de información.

Utilizamos una variable de sesión para el usuario “alumno” con la finalidad de que una vez realizado el examen no pudiera volver a resolverlo y cambiara la calificación previamente guardada en la base , terminando su sesión en el instante en que envía sus respuestas del examen que resolvió.

El profesor no interviene al momento de asignarle la calificación al alumno pues esto lo hace directamente el servidor, el cual la guarda en la base de datos y la imprime en la página web para que el alumno vea su calificación al término del examen. Por tanto el alumno sabe en ese instante la calificación que obtuvo en el examen aplicado.

El profesor sólo puede consultar información, en ningún momento tiene acceso directo con ella, el único que tiene acceso a toda la información es el administrador del sistema, con el fin de llevar un mejor control del alumnado y de sus calificaciones.

PROYECCIONES

5.- Proyecciones.

El sistema descrito en éste trabajo tiene la capacidad de crecer y ser rediseñado de acuerdo a las necesidades de los usuarios, pues la base de datos en esta ocasión solo esta diseñada para una sola materia y es la parte de éste sistema que requiere de mayor diseño por que el número de usuarios aumentaría por lo tanto los datos que se manejarían en la base serian más de los que utiliza en este momento, por lo que para poder ser utilizada en otras materias la base de datos tiene que ser diseñada de acuerdo a la magnitud de los usuarios que puedan utilizarla. Lo anterior se comenta ya que este sistema puede ser la base para que otros estudiantes puedan ampliar el sistema y diseñarlo con otros lenguajes de programación que pueden ser diferentes o iguales a los utilizados en este sistema.

5.1 Resultados esperados.

Una vez realizado la implementación del diseño del sistema se realizaron pruebas, las cuales constaron de dar de alta a los usuarios (alumnos, profesores, administradores) otorgándoles a cada uno los atributos que le corresponden dentro del sistema, después se capturaron los tipos de examen a realizar según sea el periodo que corresponda. Se realizo una prueba a los alumnos aplicando el examen.

Los resultados obtenidos fueron buenos, ya que el sistema soporto todas las consultas que realizaron los alumnos al momento de enviar su examen resuelto y obtener la calificación de éste casi instantáneamente. Una vez cerrada la sección por el alumno éste ya no podía ingresar nuevamente.

Finalmente no hubo fuga de información por lo tanto podemos decir que el sistema cumple con todas las características de ser seguro, confiable y estable.

5.2 Desarrollo propuesto.

El objetivo a futuro es la implementación de un sistema de exámenes en línea para todas las asignaturas de la Facultad de Ingeniería, el cual serviría como un examen diagnostico antes de la aplicación del examen

departamental ayudando al estudiante a detectar los temas en donde tiene bajo desempeño.

Otra aplicación propuesta sería que en la parte de aplicación del examen no solo fuera de opción múltiple si no que tuviera la posibilidad de que el alumno contestara en forma de texto a las preguntas, esto se pudiera realizar por medio de comparación de cadenas, encriptación, etc, el cual sería otro trabajo de investigación por si mismo puesto que la cantidad de información que existe para realizar este tipo de validación es muy extensa.

Este desarrollo esta propuesto para realizarse en software libre pues se puede librar de pagar licencias.

5.3. Seguridad.

Generar la seguridad en una red requiere establecer un conjunto de reglas, regulaciones y políticas que no dejan nada al azar. El primer paso para garantizar la seguridad de los datos es implementar las políticas que establecen los matices de la seguridad y ayudan al administrador y a los usuarios a actuar cuando se producen modificaciones, esperadas como no planificadas, en el desarrollo de la red.

Prevención

La mejor forma de diseñar las políticas de seguridad de los datos es optar por una perspectiva preventiva. Los datos se mantienen seguros cuando se evita el acceso no autorizado. Un sistema basado en la prevención requiere que el administrador conozca todas las herramientas y métodos disponibles que permiten mantener la seguridad de los datos.

Autenticación

Para acceder a la red, un usuario debe introducir un nombre de usuario y una contraseña válida. Dado que las contraseñas se vinculan a las cuentas de usuario, un sistema de autenticación de contraseñas constituye la primera línea de defensa frente a usuarios no autorizados.

Es importante no permitir un exceso de confianza en este proceso de autenticación engañándonos con una falsa idea de seguridad. Por ejemplo, en una red de *peer-to-peer*, casi todos los usuarios pueden entrar en el sistema con un nombre y contraseña única. Esto sólo puede proporcionar a un usuario acceso completo a la red, de forma que cualquier cosa que se comparta está disponible para este usuario. La autenticación funciona sólo

en una red basada en servidor, donde el nombre y contraseña de usuario debe ser autenticada utilizando para ello la base de datos de seguridad.

Entrenamiento

Los errores no intencionados pueden implicar fallos en la seguridad. Un usuario de red perfectamente entrenado probablemente va a causar, de forma accidental, un número menor de errores que un principiante sin ningún tipo de experiencia, que puede provocar la pérdida de un recurso dañando o eliminando datos de forma definitiva.

El administrador debería asegurar que alguien que utiliza la red esté familiarizado con sus procedimientos operativos y con las tareas relativas a la seguridad. Para lograr esto, el administrador puede desarrollar una guía breve y clara que especifique lo que necesitan conocer los usuarios y obligar a que los nuevos usuarios asistan a las clases de entrenamiento apropiadas.

Encriptación de Datos

Como sabemos, en un Sistema de Comunicación de Datos, es de vital importancia asegurar que la información viaje segura, manteniendo su autenticidad, integridad, confidencialidad y el no repudio de la misma entre otros aspectos.

Estas características solo se pueden asegurar utilizando las Técnicas de Firma Digital Encriptada y la Encriptación de Datos.

Métodos de encriptación:

Para poder Encriptar un dato, se pueden utilizar tres procesos matemáticos diferentes: Los algoritmos HASH, los simétricos y los asimétricos.

Algoritmo HASH:

Este algoritmo efectúa un cálculo matemático sobre los datos que constituyen el documento y da como resultado un número único llamado MAC. Un mismo documento dará siempre un mismo MAC.

Algoritmos Simétricos:

Utilizan una clave con la cual se encripta y desencripta el documento. Todo documento encriptado con una clave, deberá desencriptarse, en el proceso inverso, con la misma clave. Es importante destacar que la clave debería viajar con los datos, lo que hace arriesgada la operación, imposible de utilizar en ambientes donde interactúan varios interlocutores.

Algoritmos Asimétricos (RSA):

Requieren dos Claves, una Privada (única y personal, solo conocida por su dueño) y la otra llamada Pública, ambas relacionadas por una fórmula matemática compleja imposible de reproducir. El concepto de criptografía de clave pública fue introducido por Whitfield Diffie y Martin Hellman a fin de solucionar la distribución de claves secretas de los sistemas tradicionales, mediante un canal inseguro. El usuario, ingresando su PIN genera la clave Pública y Privada necesarias. La clave Pública podrá ser distribuida sin ningún inconveniente entre todos los interlocutores. La Privada deberá ser celosamente guardada.

Cuando se requiera verificar la autenticidad de un documento enviado por una persona se utiliza la Clave Pública porque el utilizó su Clave Privada.

5.4. Manual de uso del sistema.

En este apartado tratamos de prevenir el mal uso de nuestra base de datos, con la finalidad de no provocar problemas irreparables en el sistema y lograr el uso óptimo del mismo, por lo cual nos dirigimos al usuario: *administrador*, que es el que interactúa directamente con todo el sistema y la base de datos. Por tal motivo esto no es precisamente un manual, sino un conjunto de instrucciones y recomendaciones a seguir para los usuarios.

- 1.- El administrador no debe dejar espacios en blanco cuando éste capture los datos de los demás usuarios.
 - 2.- El administrador no debe escribir caracteres raros o que no coincidan con la información que el sistema requiera.
-

- 3.- El administrador debe ingresar las preguntas de los exámenes en orden.
- 4.- El administrador no debe rebasar el tamaño del enunciado ingresado en la ventana de pregunta o de respuestas.
- 5.- El administrador debe tener cuidado con la información que maneja pues si existe alguna fuga de información otro usuario puede acceder a la base de datos e incluso dañarla.

El uso adecuado del sistema puede prevenir muchas fallas y ahorrar mucho tiempo en soporte técnico y mantenimiento. Ciertamente el sistema está diseñado para no tener errores, sin embargo, muchas veces las personas que utilizan los sistemas no tienen una preparación adecuada y cometen errores que pueden ser fatales para los sistemas.

6.- Conclusiones.

Podemos concluir que el sistema funciona con éxito ya que se probó con una materia de la carrera de ingeniería mecánica y se obtuvieron resultados favorables. El desarrollo de los exámenes resultó exitoso y esperamos que se pueda utilizar de la misma forma en otras materias de la misma dependencia a la cual se elaboro el sistema descrito en este documento.

El sistema de exámenes en línea es una herramienta más que ayuda al aprendizaje del estudiante ya que funciona como una alternativa con la cual se podrá visualizar las deficiencias en temas del curso que se este tomando.

Creamos una base de datos fuerte, estable, y funcional que es donde finalmente se guardan los resultados de los alumnos.

Con lo cual el objetivo de este trabajo, el de diseñar un sistema de exámenes en línea ha sido cumplido exitosamente.

NOTAS BIBLIOGRAFICAS.

- (1) URREA Chavez Julio Cesar. Protocolos de red: Protocolo TCP/IP. México
1997. Lucas Morea / Sinexi S.A.
 - (2) URREA Chavez Julio Cesar. ¿Que es internet?. México. 1997.
Lucas
Morea / Sinexi S.A.
 - (3) WALES Jimmy. HTML. 2005. Fundación Wikimedia.
 - (4) CHISTRIAN Van Der Henst. Introducción al ASP. Abril. 2001.
Maestrosdelweb.
 - (5) SOLANO Alex. SQL Server 2000. Septiembre 2002. Net
Veloper.
 - (6) WALES Jimmy. Bases de datos. 2005. Fundación Wikipedia.
 - (7) GOOGLE Dirson. ¿Qué son las páginas dinámicas?. 2005.
Google inc.
 - (8) WALES Jimmy. Servidor HTTP Apache. 2005. Fundación
Wikipwdia.
 - (9) ROMERO Serrano Noe de Jesús. Propuesta de norma para
redes inalámbricas en México. México D.F. UNAM 2005
 - (10) C. J. DATE An introduction to database systems
Addison-Wesley publishing company
-

BIBLIOGRAFÍA y Direcciones electrónicas.

- Aprendiendo ASP. NET.

PAYNE, CHRIS. Ed. Prectice Hall. 994 pág. Mexico, 2002.

- A guide to SQL/DS

C. J. DATE with COLIN J. WHITE Addison- Wesley publishing company

- An introduction to database systems

C. J. DATE Addison-Wesley publishing company

<http://www.itlp.edu.mx/publica/tutoriales/redes/>

<http://www.monografias.com/trabajos/protocolotcpip/protocolotcpip.shtml>

<http://es.wikipedia.org/wiki/HTML>

<http://html.conclase.net/tutorial/html/>

<http://www.desarrolloweb.com/articulos/717.php?manual=15>

http://www.aspfacil.com/articulos/vbs_manual.asp

<http://www.microsoft.com/hcl/>

<http://google.dirson.com/posicionamiento.net/paginas-dinamicas/>

<http://www.desarrolloweb.com/articulos/713.php?manual=27>

http://www.videoedicion.org/manuales/varios/servidores_proxy.htm

<http://www.htmlpoint.com/iis/index.html>

http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

<http://www.desarrolloweb.com/articulos/1112.php?manual=41>

http://fmc.axarnet.es/redes/tema_10_m.htm

<http://html.rincondelvago.com/criptacion-de-datos.html>

<http://www.maestrosdelweb.com/editorial/aspintro/>
