

DEDICATORIA

A la Virgen del Perpetuo del Socorro por haberme dado el aliento la fuerza y la sabiduría y así llevar acabo mi meta de ser un Profesional en la vida.

A mi Familia especialmente a mis Padres y al Lic. José Alberto Cerda Campos, por haberme respaldado y orientarme por el camino correcto y así poder obtener el título de Lic. En Computación.

También agradezco a todo el personal Docente de la facultad de Ciencias (Carrera Computación) especialmente a los profesores que lealmente me dieron todas las nociones necesarias en la formación profesional del área Informática.

Ronald Alberto Méndez Cerda

DEDICATORIA

A Dios por brindarme la fuerza y la sabiduría necesaria para llegar hasta la meta que me propuse alcanzar.

A mis padres por brindarme su apoyo incondicional y por haber depositado en mi el anhelo de ver sus esfuerzos materializados.

A mis maestros por compartir sus conocimientos y brindarme su apoyo en las diferentes etapas de mi vida estudiantil.

Carlos Alberto Pérez López

DEDICATORIA

Dedico este trabajo monográfico primeramente a Dios quien me a guiado por el camino correcto de la sabiduría e iluminado mis pasos mostrándome la luz de la enseñanza.

A mi familia que de manera comprensible y enérgica me han apoyado en todo, momento y aún en los momentos más difíciles tanto moral como económicamente.

También agradezco a todo el personal Docente y administrativos de la facultad de Ciencias (Carrera Computación) especialmente a los profesores que lealmente me dieron todas las nociones necesarias en la formación profesional del área Informática.

Byron Ernesto Muñoz Romero

AGRADECIMIENTO

Agradecemos la realización de este proyecto monográfico a Dios primeramente, por habernos permitido la vida para ver realizado nuestros esfuerzos de cinco años.

A nuestros padres quienes nos brindaron su apoyo incondicional durante estos años de estudio.

A nuestro tutor **MSc. Francisco Danilo Padilla Contreras** quien desde el inicio de este proyecto nos brindo su apoyo con sus conocimientos.

Al cuerpo de profesores del departamento de computación quienes fueron los pilares fundamentales en nuestra formación profesional.

Atentamente:

Br. Byron Ernesto Muñoz Romero.

Br. Ronald Alberto Méndez Cerda.

Br. Carlos Alberto Pérez López.

ÍNDICE

	Páginas
INTRODUCCIÓN.....	1
JUSTIFICACIÓN.....	2
OBJETIVOS.....	3
MARCO TEÓRICO.....	4
METODOLOGÍA.....	9
RESULTADO DEL ANÁLISIS.....	38
DISEÑO	53
CONCLUSIÓN.....	73
BIBLIOGRAFÍA.....	74
ANEXOS.....	75

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA-LEON

FACULTAD DE CIENCIAS

DEPARTAMENTO DE COMPUTACION



Monografía para optar al título de Licenciado en Computación

“Conexión de Base de Datos con la WEB utilizando ASP”

PRESENTADO POR:

Br. Byron Ernesto Muñoz Romero

Br. Ronald Alberto Méndez Cerda

Br. Carlos Alberto Pérez López

TUTOR:

MSc. Francisco Danilo Padilla Contreras.

León, Octubre de 2003



INTRODUCCIÓN

La Internet es la red más grande que se ha llegado a desarrollar, constituye una fuente de información muy amplia propósito por el cual fue creada, jugando un papel muy importante para localizar / enviar / recibir información de diversos tipos.

Se conoce que alrededor de 25 millones de usuarios se conectan a diario a la Internet con el objetivo de satisfacer sus diferentes necesidades de información.

Es una herramienta muy importante que nos permite acceder a mucha información de diferente índole como por ejemplo: aspectos de programación, socios económicos, políticas, culturales, venta y compra de productos, deportes, música, revistas, correo electrónico, etc.

La presente aplicación nos muestra un estudio generalizado del ASP, la cual es una herramienta incluida en los componentes del IIS que incorpora Scripts de Visual Basic, Java y Perl. La tecnología ASP ha sido diseñada por Microsoft para facilitar la creación de sitios Web con gran sencillez y eficiencia debido al uso de Visual Basic Script.

Nuestro sistema ejemplo tendrá énfasis en demostrar el acceso a Base De Datos desde Internet haciendo uso de páginas ASP para el acceso a dicha Base De Datos, y del gestor de Base De Datos (Microsoft Access).

Históricamente las Bases De Datos eran utilizadas en instituciones o en redes locales, en la actualidad el Web permite acceder a Bases De Datos desde cualquier parte del mundo. Ofreciendo un manejo dinámico y una gran flexibilidad de los datos que es una gran ventaja que no poseen otros medios informáticos.



JUSTIFICACIÓN

En tiempos remotos la programación no poseía un auge tan influyente como en la actualidad, además de que no disponían de herramientas muy sofisticadas para el desarrollo de aplicaciones Web que fueran fáciles de interpretar y de un manejo sencillo de sus componentes y herramientas, también no se disponía de máquinas tan poderosas como en estos días.

En nuestra aplicación presentamos los diferentes componentes que se conjugan para crear una aplicación que utilice programación de páginas ASP, que aunque no es nada novedoso, es una herramienta que nos permite crear páginas activas dentro de un servidor Web que en tiempos pasados se hablaba solo de páginas estáticas, y además presentar la gran sencillez y eficiencia para la conexión de Bases De Datos con la Web, independientemente del generador de Base De Datos que se utilice.



OBJETIVOS

Generales:

- Desarrollar una aplicación de demostración que nos presente la manera en que pasa la información del Web hacia la Base De Datos y viceversa haciendo uso de páginas ASP.

Específicos:

- Elaborar un ejemplo de páginas web, con uso de ASP.
- Conectar una Base De Datos diseñada en Microsoft Access A un sitio Web utilizando ASP.
- Presentar el mecanismo de funcionamiento de la tecnología ASP.



MARCO TEÓRICO

❖ ¿Qué es Internet y cómo funciona?

Es una colección global de redes de computadoras conectando miles de lugares y millones de usuarios ubicados en todo el mundo con el uso de la tecnología TCP/IP.

❖ **Servidor** : Ordenador que posee un alto rendimiento y gran capacidad de almacenamiento y a través de un sistema operativo y una red administra y controla otros ordenadores denominados clientes.

❖ **Cliente** : Conjunto de ordenadores que mediante un medio de comunicación acceden al servidor con el fin de cumplir algunos propósitos (En algunos casos se deberá de entender como cliente a un Navegador de Internet).

❖ **Cliente / servidor** : Son los ordenadores que pueden compartir recursos de sistemas, unidos por medio de una topología de Red.

❖ ¿Qué es ASP?

Active Server Pages (Páginas Activas en el Servidor) es un entorno de programación que provee la habilidad de combinar código HTML y componentes ActiveX del servidor para crear poderosas aplicaciones para Internet que se ejecutan en el servidor, es una herramienta incluida en los componentes de IIS que incorpora scripts de Visual Basic, Java y Perl, los que se ejecutan en el servidor para la creación de Páginas Web dinámicas. La tecnología ASP ha sido diseñada por Microsoft para facilitar la creación de sitios Web con gran sencillez y eficiencia debido al uso de Visual Script.



❖ ¿Cómo trabaja ASP?

Al acceder una página ASP esto es lo que sucede:

- 1- El navegador solicita el archivo ASP desde el servidor Web.
- 2- El Script del lado del servidor comienza a ejecutarse con el código ASP.
- 3- ASP procesa el archivo solicitado de manera secuencial, ejecuta algunos comandos contenidos en el archivo, y produce una página web en formato HTML.
- 4- La página Web es enviada al navegador.

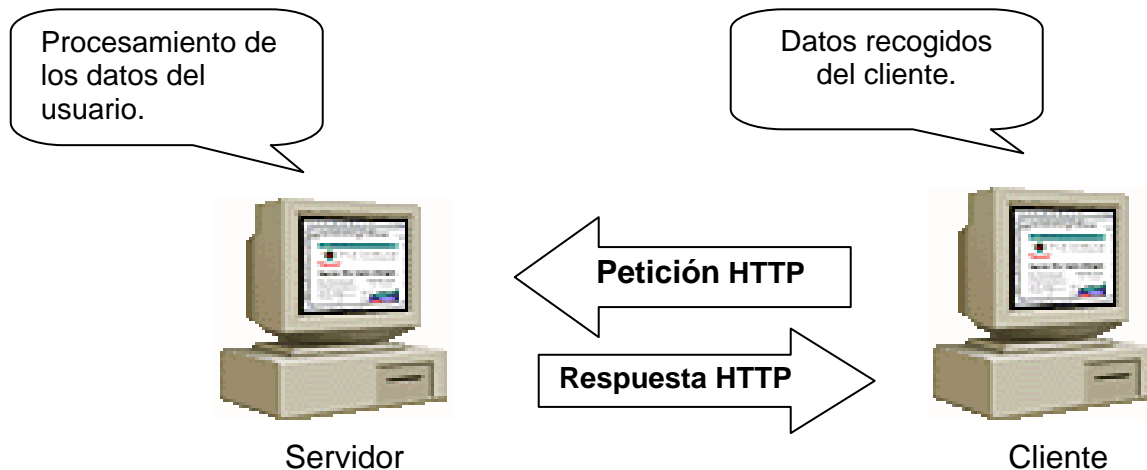


Figura 1.

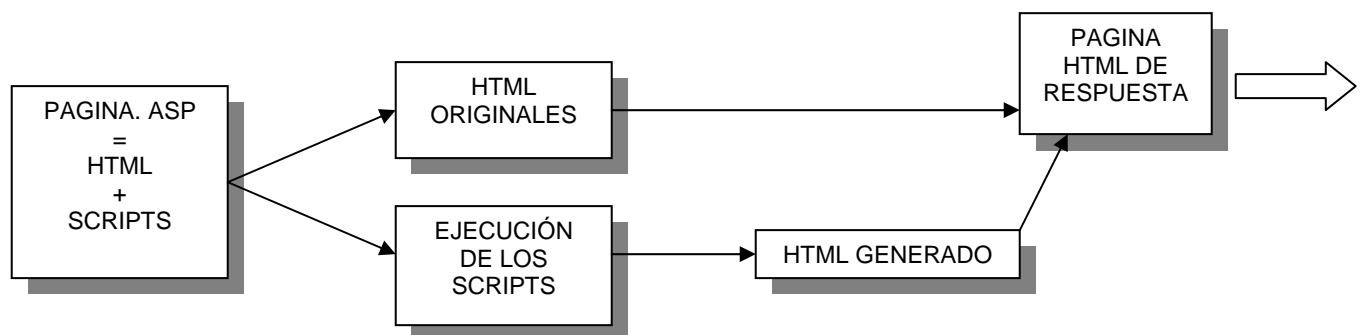


Figura 2.

Figura 1 y 2 Muestra la manera de funcionamiento del ASP.



❖ ¿Qué es Bases De Datos?

Cuando nos referimos a Bases De Datos en un sistema este consiste en almacenar la información en varias tablas de forma eficiente, y luego para sacar esta información se requiere de un lenguaje de manipulación de datos.

❖ HTML(Hyper Text Markup Language)

Tiene como principio esencial de lenguaje el uso de las Etiquetas (TAGS), funciona de la siguiente manera :

< xxx > Este es el inicio de una etiqueta.

</ xxx > Este es el cierre de una etiqueta.

Las letras de la etiqueta pueden estar en mayúsculas o minúsculas indiferentemente.

Este es un lenguaje para la programación de sitios web.

❖ **ADO:** ActiveX Data Objects (Objetos de Datos ActiveX) Usa una tecnología ampliable y de fácil uso para agregar acceso a Bases De Datos desde páginas Web. Es utilizado para escribir secuencia de comandos compactos y escalables que conecten con Bases De Datos compatibles con ODBC. ADO se incluye con Microsoft Internet Información Server y con Microsoft Visual InterDev.

❖ **DBMS:** Data Base Management System (Sistemas de Manejo de Bases De Datos) Sistemas generadores de Bases De Datos, se ocupan de acceder y actualizar las Bases De Datos, tales como Access, FoxPro, Oracle, SQL Server etc.



❖ **ODBC:** Open Data Base Connetivity (Conectividad Abierta a Bases de Datos)

Es una interfaz que permite a las aplicaciones acceder a distintos tipos de Bases De Datos desde diversas especificaciones de orígenes de datos. Para ello se necesitan los diferentes controladores ODBC con los que se obtiene información de los orígenes de datos.

❖ **IIS:** Internet Information Server (Servidor de Información de Internet), es el software estándar que soporta comunicaciones en Internet sobre el Sistema Operativo Windows NT, incluye el servicio de publicación de aplicaciones Web (Servicio WWW) que es el más importante, el servicio FTP, servicio de correo electrónico, servicio de noticias, programación de páginas activas en el servidor ASP, acceso a Bases De Datos, etc.

❖ **PWS:** Personal Web Server (Servidor Web Personal), es la versión pequeña del IIS 4 que se puede instalar sobre Windows 95/98 o Windows NT Workstation 4, su funcionalidad es menor que la de IIS al igual que el numero de componentes que soporta es menor. Lo utilizaremos para realizar las respectivas pruebas de nuestra aplicación.

❖ **ActiveX:** Este es un término que engloba tecnologías de Microsoft que permiten a los programadores crear contenido interactivo para World Wide Web. Un conjunto de tecnologías de interoperabilidad independientes del lenguaje que permiten que los componentes de software escritos en diferentes lenguajes funcionen juntos en entornos de red. Los elementos fundamentales de la tecnología ActiveX son el Modelo de objetos componentes (COM) y el Modelo de objetos componentes distribuido (DCOM).



❖ **COM:** Components Objects Model (Modelo de Objetos Componentes). Es el modelo de programación orientada a objetos que define cómo interactúan los objetos con una única aplicación o entre distintas aplicaciones. En COM, el software cliente tiene acceso a un objeto mediante un puntero a una interfaz (un conjunto de funciones relacionadas llamadas métodos) del objeto.

❖ **DCOM :** Modelo de objetos componentes distribuido. Extensiones del Modelo de objetos componentes (COM) que facilita la distribución transparente de objetos a través de redes y de Internet. DCOM forma parte de la especificación administrada por The Open Group para la distribución en plataformas heterogéneas.



METODOLOGIA

La presente aplicación ilustrará la conveniencia para el intercambio de información entre los usuarios y el servidor de páginas Web, utilizando el concepto de Páginas Web activas o dinámicas que el modelo ASP (Active Server Pages) soporta.

La capacidad de procesamiento que sustenta las páginas dinámicas se puede llevar a cabo utilizando alguno de los siguientes modelos:

Procesamiento en el equipo del usuario = Páginas Activas en el cliente.

Procesamiento en el equipo donde reside el servidor del Web = Páginas Activas en el servidor.

Procesamiento mixto = Páginas Activas en el cliente + Páginas Activas en el servidor.

La ventaja principal de las páginas activas en el cliente es la descarga de trabajo que le proporciona al equipo servidor la posibilidad de traspasar cálculos a los usuarios. Otra importante ventaja se basa en el ahorro de comunicaciones (ancho de banda) que se pueden experimentar en muchas aplicaciones que ejecutan procesos en el cliente, y evitan de esta manera realizar continuos traspasos de información con el servidor Web.

En nuestro caso en particular profundizaremos en el uso del modelo de páginas activas en el servidor, que proporciona un adecuado soporte de procesamiento, a la vez que permite una coexistencia con páginas activas en el cliente, se pueden realizar accesos a componentes ActiveX que se ejecutan en el servidor, generándonos un acceso a distintos tipos de Bases De Datos sencillo y eficiente a través de ODBC.

**Microsoft Access 2000.****→ Ventajas:**

Enfocado a las Bases de Datos: Es un sistema administrador de Bases De Datos relacionales, por lo que es apropiado para resolver problemas de dicha naturaleza.

Amigable: Brinda muchas facilidades al programador, ya que, es fácil de manejar y programar.

Generador de código: Gran parte del código es generado automáticamente por el propio sistema.

Seguridad: Restringe el acceso a la Base De Datos por parte de usuarios no autorizados al establecer niveles de seguridad a través de la creación de usuarios y grupos.

Costo: El costo de la licencia y del acceso a documentación técnica es baja.

Interoperabilidad: Se puede crear un enlace del archivo de Base De Datos a una interfaz hecha en Visual Basic, Fox Pro, Dbase, Paradox y otros.

Soporte para publicar en el Web: Permite crear páginas Web dinámicas para realizar consultas sobre servidores Web tomando datos de una Base De Datos Access.

Integridad Referencial: Resuelve la integridad referencial mediante reglas a nivel de campos y a nivel de registros definidos por el usuario. Estas reglas rigen las relaciones entre las claves principales y externas de las tablas, y son:

1. **Insertión de datos:** No se permitirá agregar registros a una tabla relacionada si no existe ningún registro asociado en la tabla primaria.



2. **Actualización de datos:** No se permitirá cambiar los valores de una tabla primaria cuando tales cambios supongan dejar registros huérfanos en una tabla relacionada.
3. **Eliminación de datos:** No se permitirá eliminar registros de una tabla primaria cuando tenga registros relacionados coincidentes.

→ **Desventajas:**

Manejo: La interfaz usuario es mas difícil de elaborar, con respecto a otros Sistemas manejadores de Bases De Datos y otras herramientas de desarrollo como Visual Basic.

Integridad: La integridad de datos es menos eficiente con respecto a otros, como Oracle.

Rendimiento: No esta capacitado para manejar un volumen de información muy amplio, pues el rendimiento ira desminuyendo. La Base De Datos no deberá ser mayor a un 1Gbde almacenamiento.

Diseño De La Base De Datos.

Para tal efecto hay que proceder de una manera selectiva y respaldado de un estudio amplio y categorico de cómo estará constituida la Base De Datos a utilizar, teniendo en cuenta que trabajaremos con el administrador de Bases De Datos Microsoft® Access, y que nuestra aplicación es solamente de demostración y por ende nuestra Base De Datos es de ejemplo.

Al momento de diseñar la Base De Datos debemos tener en cuenta algunos inconvenientes tales como:



1. El conocimiento previo por parte de los usuarios de la ubicación de los datos.
2. Las Bases De Datos pueden contener muchas tablas, cada una de ellas conteniendo varios campos o columnas.
3. Nombres de tablas y campos son a menudo extraños y llevan a la confusión a los usuarios.

Estas complejidades son ocultas al usuario final, y la solución a este tipo de problemas es implementar una Base De Datos bien diseñada.

Personal Web Server 4.0

En el paquete de Personal Web Server, Microsoft ha unido nuevas funcionalidades que hacen de Microsoft® Windows 98® una plataforma aún más fácil de utilizar para compartir información en un sitio Web personal.

El paquete de Personal Web Server hace fácil instalar cualquiera de las siguientes nuevas características:

✓ Microsoft Personal Web Server 4.0

Un servidor Web de escritorio que se puede utilizar para alojar un sitio Web en la intranet corporativa o para desarrollar y probar un sitio Web antes de alojarlo en un proveedor de servicios Internet (ISP).

✓ Extensiones de servidor de Microsoft FrontPage

Permite el uso de Microsoft® FrontPage® para administrar el sitio Web, así como crear el contenido del sitio.

✓ Microsoft Transaction Server 2.0

Permite la creación de aplicaciones de Microsoft® Transaction Server (MTS). Las transacciones son operaciones del servidor que se ejecutan correctamente o fallan



en su totalidad, incluso aunque las operaciones consten de varios pasos. MTS también admite el aislamiento del proceso de las aplicaciones.

✓ **Microsoft Data Access Components 1.5**

Utilización sencilla de bases de datos con compatibilidad para objetos de datos ActiveX™ (ADO) y el controlador Microsoft® Access.

✓ **Message Queue Server 1.0**

Facilita a los programas de aplicación la comunicación rápida, confiable y asíncrona con otros programas de aplicación mediante el envío y la recepción de mensajes.

Las características claves de Message Queue Server (MSMQ), como la compatibilidad de ActiveX, amplios controles de seguridad, eficaces herramientas de administración un extenso conjunto de características y la integración con productos estratégicos de Microsoft como Internet Information.

Server y MTS, hacen de MSMQ el producto de cola de mensajes preferente para aplicaciones que se ejecutan en Windows 95, Windows 98 y Windows NT. El paquete de Personal Web Server incluye los clientes dependientes e independientes de MSMQ.

Comparación de productos: PWS para Windows 98 y PWS para Windows NT Workstation.

PWS para Windows 98 proporciona una interfaz gráfica de administración diseñada para usuarios sin experiencia previa en la creación y administración de sitios Web. Esta interfaz se proporciona también como la herramienta predeterminada de administración en Personal Web Server para Microsoft® Windows NT® Workstation; sin embargo, con la versión de Windows NT® Workstation, también puede administrar su sitio personal de



publicación con el Administrador de servicios Internet, la misma herramienta de administración con características completas que se utiliza para controlar Microsoft Internet Information Server.

**Comparación de productos**

Característica	PWS para Windows NT Workstation	PWS para Windows 98
Utilizaciones típicas	Desarrollo de sitios con características completas o publicación personal en una intranet corporativa	Publicación personal en una intranet corporativa de pequeño volumen
Servicio WWW	Sí	Sí
Servicio FTP	Sí	No
Límite de conexiones	10	10
Páginas Active Server	Sí	Sí
Registro de utilización del sitio	Formato de archivo de registro NCSA (predeterminado) MSCSV estándar y extendido (opcionales)	Formato de archivo de registro NCSA
Orígenes de publicación	Unidades locales y de red	Sólo unidades locales
Interfaz ISM opcional	Sí	No
Autenticación	Básica o Desafío / respuesta de Windows NT	Ninguna

**Instalar PWS 4.0**

Estos son los requisitos del sistema para la instalación de PWS.

Componente de hardware	Requisito	Recomendación
Procesador	486 a 33 MHz	Pentium® a 90 MHz
RAM	16 MB	20 - 32 MB
Espacio libre en disco duro	30 MB	40 MB
Monitor	VGA	Super VGA

En Windows 98, PWS ofrece tres opciones de instalación:

- **Instalación mínima** Los componentes mínimos necesarios para ejecutar PWS.
- **Instalación típica** Las opciones mínimas con funcionalidad y documentación adicionales.
- **Instalación personalizada** Presenta todos los componentes posibles como opciones, con todas las opciones incluidas en las instalaciones mínima y típica preseleccionadas.

**Para instalar Personal Web Server**

El CD de Windows® 98 incluye Personal Web Server (PWS); Sin embargo, PWS requiere un paso independiente de instalación para estar completamente instalado y configurado. Para instalar PWS, necesita el CD de Windows 98 y dar los pasos siguientes:

1. Inserte el CD de Windows 98 CD en la unidad.
2. Haga clic en **Inicio** y, a continuación, haga clic en **Ejecutar**.
3. En el cuadro de diálogo **Ejecutar**, escriba **x:\add-ons\pws\setup.exe**.
4. Substituya x por la letra de la unidad de CD y haga clic en **Aceptar**.
5. Siga las instrucciones en la instalación de Personal Web Server.

Limitaciones y problemas conocidos

La siguiente lista contiene problemas y limitaciones que se sabe que existen en esta versión:

- Personal Web Server no es compatible con servicios FTP.
- Si actualiza desde una versión anterior de PWS y tiene problemas para ver Default.asp, es probable que tenga una versión más antigua de este archivo. Para instalar la nueva versión PWS 4.0 de este archivo, debe quitar manualmente la versión antigua de Default.asp. La ubicación es C:\InetPub\Wwwroot.
- En algunos casos, los archivos que quedan tras una instalación anterior de las extensiones de servidor de FrontPage pueden evitar que las nuevas versiones de este componente se instalen correctamente. Si el programa de instalación de PWS informa de errores durante las Extensiones de servidor de FrontPage o si PWS, Microsoft® FrontPage o Microsoft® Visual InterDev fallan al conectarse a un Web de FrontPage, utilice el siguiente procedimiento para corregir el problema:



1. Ejecute el programa de instalación de PWS, elija **Agregar o quitar** y desinstale las **Extensiones de servidor de FrontPage**.
 2. Elimine los siguientes archivos del directorio \Windows\System\:
Fp30txt.dll, Fp30utl.dll y Fp30wel.dll.
 3. Elimine el directorio \Archivos de programa\MicrosoftFrontPage\version3.0 incluidos todos los archivos y subdirectorios.
 4. Ejecute el programa de instalación de PWS y reinstale las **Extensiones de servidor de FrontPage**.
- Ejecutar Personal Web Server 4.0 en un equipo con un procesador Intel 486 de velocidad inferior a 50 MHz y con 16 MB o menos de RAM aumenta la probabilidad de que los motores de secuencias de comandos se degraden a un *modo de empaquetado de pilas de memoria*, un proceso que evitará que el servidor acepte nuevas conexiones. El proceso de empaquetado de pilas puede tardar de 20 a 30 minutos en completarse (dependiendo de la velocidad del microprocesador).
- **Páginas Active Server**
- Si la única Base De Datos que utiliza con las páginas Active Server® (ASP) es Microsoft SQL Server o sea, nunca utiliza archivos de Microsoft® Access (.mdb) con ASP y crea instancias del componente Data Base Access que almacena en el objeto **Session**, puede mejorar el rendimiento si cambia en el Registro el modelo de subprocesamiento para las instancias del componente Data Base Access de Controlado a Ambos. Para cambiar de Controlado a Ambos, haga doble clic en Makefre15.bat en la carpeta Archivos de programa\Archivos comunes\System\Ado.



Para invertir este proceso, (o sea, para cambiar el modelo de subprocesamiento de Ambos a Controlado) haga doble clic en Makeapt15.bat en la carpeta Archivos de programa\Archivos comunes\System\Ado.

Nota: Recomendamos que *no* almacene instancias del componente Database Access en el objeto **Session**.

- En la documentación de ASP se indica incorrectamente que el método MapPath no es compatible con la sintaxis de rutas de acceso relativas, como */ruta de acceso*. El método MapPath admite sintaxis de rutas de acceso relativas si la propiedad EnableParentPaths de la metabase está habilitada. Cuando la propiedad EnableParentPaths no está habilitada, utilizar el método MapPath con una ruta de acceso relativa produce un error.
- El objeto Scripting.Dictionary está marcado erróneamente como procesamiento Ambos. Se debería marcar como subprocesamiento Controlado. Para cambiar esto, utilice el Editor del Registro para abrir la siguiente clave del Registro:

```
HKEY_CLASSES_ROOT\
CLSID
\{EE09B103-97E0-11CF-978F-00A02463E06F}
\InprocServer32
```

Cambie el valor con nombre para ThreadingModel a Apartment. Si utiliza el objeto Dictionary con alcance de aplicación sin hacer este cambio, pueden dañarse los datos.

- Las propiedades establecidas para los objetos Request, Response y Server de ASP sólo son válidas durante el proceso de una petición. Por ejemplo, si establece una propiedad en Global.asa, como Response.Buffer = True, la propiedad sólo se aplicará a una única petición y no a todas las peticiones.



- Actualmente hay problemas conocidos al utilizar PerlScript con ASP. Si ejecuta PerlScript de ActiveWare, póngase en contacto con ActiveWare para obtener un nuevo motor de PerlScript.
- Si crea o utiliza componentes desarrollados en Microsoft® Visual Basic5.0™ (Edición empresarial o Edición profesional) para utilizarlos con ASP, se recomienda encarecidamente que actualice a Visual Studio97 Service Pack 2, tanto para los equipos de programación como para los servidores.
- Sólo puede publicar archivos del disco duro local. Por razones de seguridad, el asistente para publicar de Personal Web Manager no publicará un archivo que resida en una unidad de red.
- En el Asistente para página principal de Personal Web Manager, los gráficos de la hoja de estilos asociada a las plantillas de página principal se ven mejor con Internet Explorer 4.01. Las mínimas diferencias entre la forma en que aparece la plantilla de página principal en IE 3.0 y en Netscape son normales.
- Microsoft® Internet Explorer versiones 3.0 y 4.0 pasa por alto los caracteres de salto de línea en campos de texto. Para agregar saltos de línea a una entrada de un campo de texto, agregue la etiqueta HTML
 al final de cada línea.
- El servicio Remote Data (RDS) 1.5 no funciona cuando está instalado como servidor en Windows 95 o Windows 98 Beta que utiliza Personal Web Server. Esto se solucionará en una versión posterior del servicio Remote Data. Actualmente, RDS se ejecuta correctamente desde Microsoft® Windows NT Server o Workstation, en plataformas x86 o Alpha.



Solución de problemas

Si tiene problemas al instalar PWS o de funcionamiento del programa tras la instalación, consulte los elementos siguientes:

1. Desinstalar versiones anteriores.

Esta versión realiza automáticamente la actualización a PWS versión 1.0. Si tiene PWS 4.0 versión Alpha o Beta 2, debe desinstalarla antes de instalar esta versión. Realice la desinstalación mediante la opción **Quitar todo** del programa de instalación incluido en el disco compacto de la versión anterior. Si la opción **Quitar todo** no está disponible, utilice **Agregar o quitar** y desactive todas las casillas de verificación.

2. Carpeta de instalación.

De manera predeterminada, Personal Web Server se instala en C:\Windows\System\Inetsrv. No puede cambiar la carpeta predeterminada de instalación. El directorio de publicación se instala, de manera predeterminada, en C:\Inetpub\Wwwroot. Si cambia el directorio predeterminado de publicación, asegúrese de que escribe un nombre de ruta completo, incluyendo la letra de unidad. El programa de instalación puede malinterpretar las rutas relativas y las que no contienen una letra de unidad.

**Probar la instalación.**

Puede probar la instalación si utiliza Internet Explorer para ver los archivos de su directorio particular.

✓ Para probar un sitio Web de su Intranet.

1. Compruebe que el equipo tiene una conexión activa de red y que el servicio Servidor WINS (u otro método de resolución de nombres) está funcionando.
2. Inicie un explorador de Web, como Internet Explorer.
3. Escriba el Localizador de recursos universal (dirección URL) para el directorio particular del nuevo servidor y presione ENTRAR.

La dirección URL es “http://” seguida del nombre de su servidor en la red de Windows y la ruta del archivo que desea ver (incluya las barras diagonales). Por ejemplo, si su sitio Web está registrado en el servidor WINS como “Admin1” y desea ver el archivo Homepage.htm en la raíz del directorio particular, debe escribir en la **Barra de direcciones** del explorador **http://admin1/homepage.htm** y, después, presionar ENTRAR. Se mostrará la página principal en la pantalla.

✓ Para probar un sitio Web que no está conectado a una red.

1. Utilice el Explorador de Windows para buscar el archivo Hosts.sam en la carpeta \Windows y ábralo con un editor de textos, como Bloc de notas.
2. Bajo la línea 127.0.0.1 localhost, escriba **127.0.0.1 nombre_de_su_equipo**, donde *nombre_de_su_equipo* es el nombre de equipo mostrado en la página de propiedades **Identificación** en el subprograma Red del Panel de control.
3. Guarde el archivo con el nombre Hosts, sin extensión.
4. Abra Internet Explorer, haga clic en **Ver, Opciones, Conexión** y seleccione **Usar un servidor proxy para acceder a Internet**.

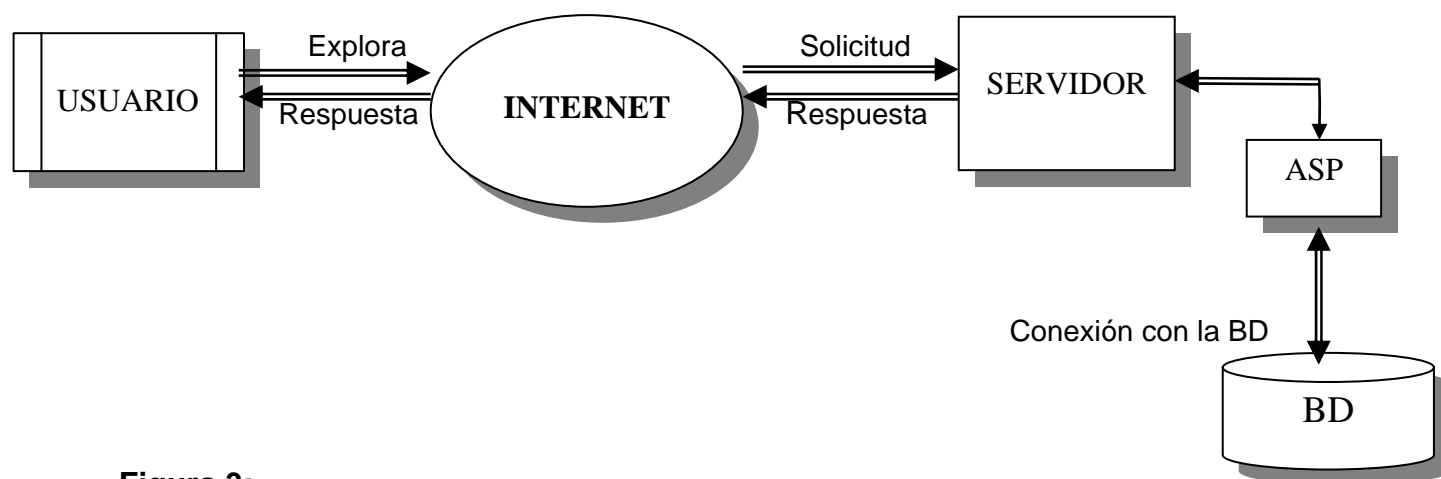


5. Haga clic en **Opciones avanzadas** y agregue el nombre de su equipo a la lista **Excepciones**.
6. Escriba **localhost** o el nombre del equipo en la **Barra de direcciones** del explorador y presione ENTRAR.

Nota: También puede utilizar la dirección IP 127.0.0.1 para llegar al sitio Web.

Ver su página principal

Si ya tiene una página default.htm en el directorio Inetpub\Wwwroot cuando instale PWS, el programa de instalación no sobrescribirá esta página. Para ver la página creada con el Asistente de página principal de PWS, haga clic en el icono Avanzadas del Administrador de Web personal y cambie el orden de los documentos en el cuadro **Habilitar documento predeterminado** a **default.asp**, **default.htm**. Si ya tiene una página default.asp en el directorio Inetpub\Wwwroot cuando instale PWS, mueva, elimine o cambie el nombre del archivo y, después, copie el archivo Inetpub\Wwwroot\lissamples\Homepage\default.asp al directorio Inetpub\Wwwroot. Después de copiar el archivo, vaya al Administrador de Web personal y haga clic en el icono Sitio Web para crear su página principal.

**MODELO CLIENTE SERVIDOR****Figura 3:**

En este diagrama se demuestra la gran importancia del servidor cuando se tiene conexión a Internet o una Intranet, debido a que el servidor es el elemento central del sistema, a el acceden los clientes por medio de Internet o la Intranet en la tecnología ASP que establece a su vez una conexión a la Base De Datos, y hace uso de diferentes componentes en conjunto con el HTML para mostrar páginas generadas dinámicamente.

NOTA: Es importante mencionar que nosotros no estamos desarrollando una aplicación que dispondrá de un servidor Web, por lo que el diagrama anterior nos sirve para una representación del funcionamiento de las aplicaciones que utilizan servidores Web que interaccionan con páginas dinámicas dentro de él mismo, las cuales pueden ser generadas con las herramientas de la tecnología ASP.



LA SEGURIDAD

Este es un aspecto en el que hay que tener mucho énfasis en todo tipo de sistema, y más en las aplicaciones Web, cuando estas interactúan con Bases De Datos evitando de esta forma el estropeamiento en conjunto de la aplicación o del sistema en particular.

La implementación de esta característica en una aplicación es de gran importancia para la interconexión del Web con Bases De Datos. En redes locales (LAN) se puede permitir o impedir a un grupo de usuarios el acceso a cierta cantidad de información, pero en la red universal Internet se ha de disponer de controles que generen un mayor rendimiento y flexibilidad para el control de manipulaciones de datos, modificación de códigos, copia de datos, etc.

Una forma de conservar la seguridad es la identificación de los usuarios, definiendo a cada grupo de usuarios un control de acceso a diferentes campos de una Base De Datos.

Para el acceso a Bases De Datos por medio de Internet es necesario disponer de una interfaz para las conexiones, la cual extraerá la información de las Bases De Datos y este a su vez le dará un formato adecuado para poder ser vista desde un Browser o navegador de Internet.

Si deseamos hacer una administración de los servicios de red que ofrece un sitio Web deben de ser configuradas las características relacionadas con la seguridad del sitio Web proporcionadas por IIS (Internet Information Server), esto en el caso de que el servidor Web posea Microsoft® Windows® NT Server 4.0 .

**PROGRAMACIÓN EN ASP**

Se conoce que anteriormente las páginas Web eran estáticas con respecto a que los usuarios únicamente realizaban el proceso de visualización de sus contenidos (HTML) en el navegador Web del cliente, el uso de estas se debe a que su creación es sencilla y fácil de instalar y mantener y además que son la base necesaria para la presentación de datos en variedad de situaciones.

Pero con la necesidad del intercambio de información entre los usuarios y el Servidor de páginas Web, con el objetivo de ofrecer un servicio mejorado al cliente se introduce el concepto de páginas Web activas o dinámicas que el modelo ASP soporta.

El procesamiento de páginas Web dinámicas deben de seguir alguno de estos modelos:

- Procesamiento en el equipo del usuario: Páginas Activas en el cliente.
- Procesamiento en el equipo donde reside el Servidor Web: Páginas Activas en el Servidor.
- Procesamiento mixto: Páginas Activas en el cliente más Páginas Activas en el Servidor.

En ASP todas las páginas Web pueden ser diseñadas con editores de HTML, puesto que las instrucciones ejecutables y el código HTML están suficientemente delimitados. El principio de la tecnología ASP es el VBScript, también pueden ser utilizados el Perl, Jscript, etc.

El ASP es una tecnología dinámica funcionando principalmente del lado del Servidor, lo cual cuando el usuario solicita un documento ASP, se ejecutan las instrucciones dentro del Script y se envía al navegador únicamente el código HTML resultante (Ver Figura 1 y 2). También permite la creación de páginas Web Activas tanto en el cliente como en el Servidor según los objetivos que se persigan.



ASP está ligado a IIS de Microsoft® Windows NT, por lo que para la ejecución de las páginas se requiere de un Servidor Microsoft® Windows NT, lo cual permite que puedan ser probadas localmente en una PC con sistemas operativos de Microsoft®.

La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que el código se encuentra únicamente archivado en el servidor, el cual al ser solicitado por medio del Web, se ejecuta, y se envía la página resultante al navegador que es lo único a lo cual el usuario tiene acceso.

ASP es una tecnología con multitareas ya que puede usarse en variedad de servicios, tales como: Acceso a Bases De Datos, envío de correo electrónico, creación dinámica de gráficos, entre otros. Por lo que muchas cosas que podemos realizar utilizando CGI pueden ser realizadas con esta tecnología, de una manera eficientemente y sencilla que es una ventaja del ASP y funcionando como un servicio dentro del servidor, o sea como un servicio que ofrece el servidor.

ASP Y EL ACCESO A BASES DE DATOS

El acceso a las Bases De Datos es una de las más eficientes características que nos ofrece la tecnología ASP, previéndonos una facilidad en el uso de las Bases De Datos. Haciendo uso de ActiveX Data Objects (ADO) y sus componentes, y ODBC(Open Data Base Connetivity), logrando con estos que los datos sean encapsulados en un objeto, lo que permite que al usuario solamente se le proyecte una imagen de los datos y no los datos mismos como ocurre con otros sistemas, ofreciéndonos así una manera sencilla y uniforme de acceso a una Base De Datos independientemente del tipo de esta que estemos usando, ya sea Microsoft® Access, SQL Server, Oracle, etc.



Para hacer interactuar la página Web con la Base De Datos, se necesita:

1. Crear la Base De Datos que albergará los datos a mostrar en las consultas a la página Web.
2. Dar de alta a la Base De Datos en el servidor para que pueda ser accedida desde la página Web.

Para esto se utiliza la interfaz Open DataBase Connectivity (ODBC), la cual sirve de enlace entre la Base De Datos y la página Web. Para acceder a los tipos de Base De Datos, se necesita los diferentes controladores ODBC, con los que se obtiene información de los orígenes de datos.

Orígenes de datos se refiere a los controladores de Bases De Datos, éstos son programas que pasan información desde una aplicación Web a una Base De Datos.

Los controladores de Bases De Datos utilizan un nombre de Origen de Datos (DSN) para encontrar e identificar una Base De Datos en particular, el DSN contiene información de configuración de la Base De Datos, seguridad de usuarios y ubicación física.

Tipos de DSN:

- a) Usuario.
- b) Archivo.
- c) Sistema.

Los DSN de Usuario y de Sistema residen en el registro del sistema de Windows. Los DSN de Sistema permiten que todos los usuarios que han iniciado una sesión en un servidor concreto tengan acceso a una Base De Datos, tienen la ventaja de poder ser accedidos por cualquier usuario, por lo que son empleados



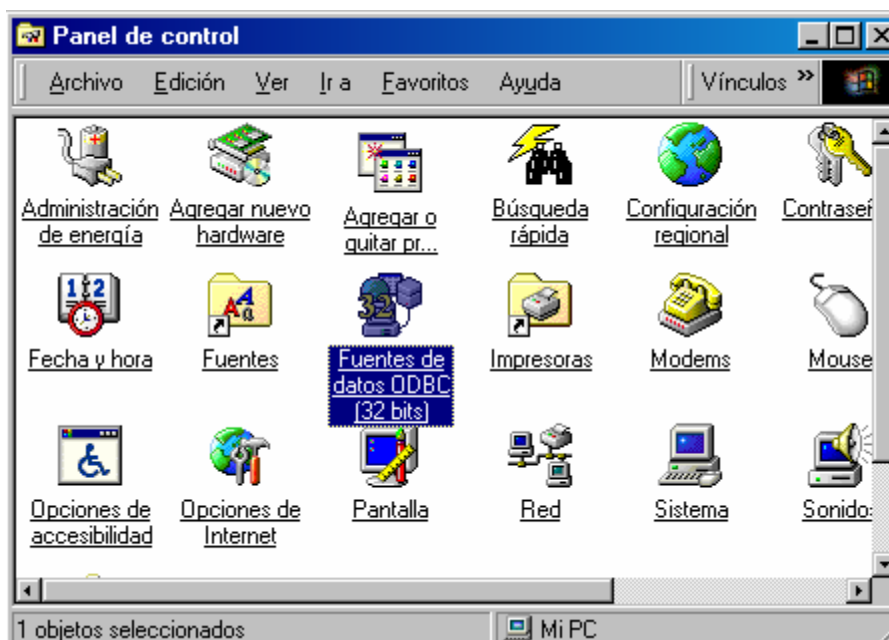
mayoritariamente por las aplicaciones Web que implementan páginas Activas en el Servidor.

Los DSN de Usuario limitan la conectividad con la Base De Datos a los usuarios que tengan los permisos de seguridad apropiados.

Los DSN de Archivos tienen la forma de archivos de texto con la extensión .dsn y se guardan en la ruta c:\Archivos de Programa\Archivos Comunes\odbc\Data Sources, proporcionan acceso a varios usuarios y son fácilmente transferibles entre un servidor y otro mediante la copia de los archivos DSN.

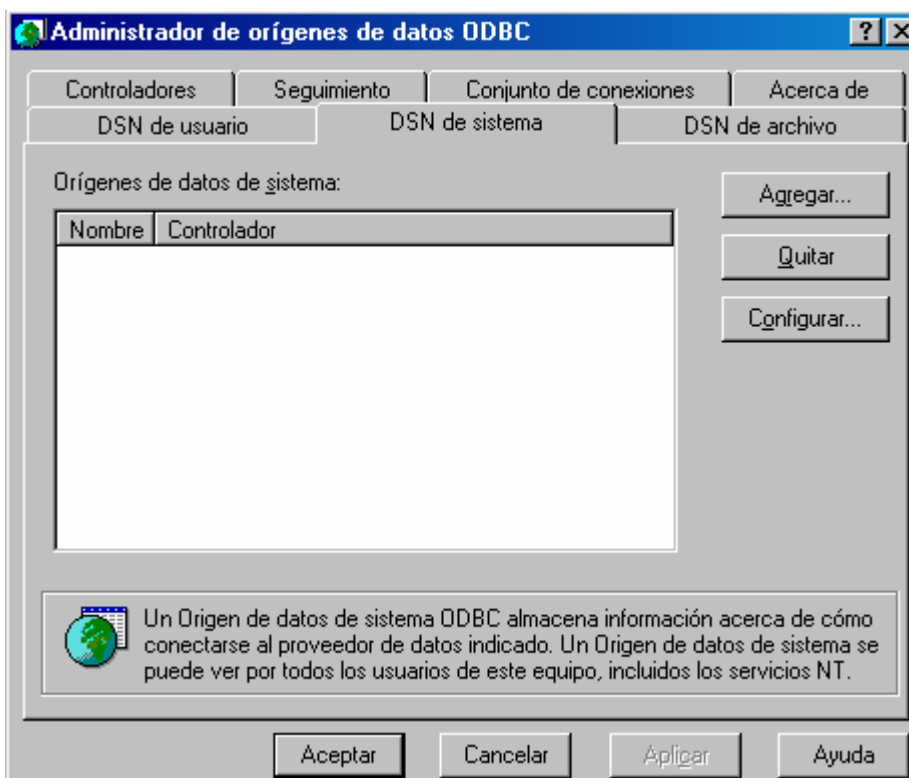
Crear un DSN de Sistema:

- a) Entrar al panel de control y hacer un doble clic en el Icono Fuentes de Datos ODBC.



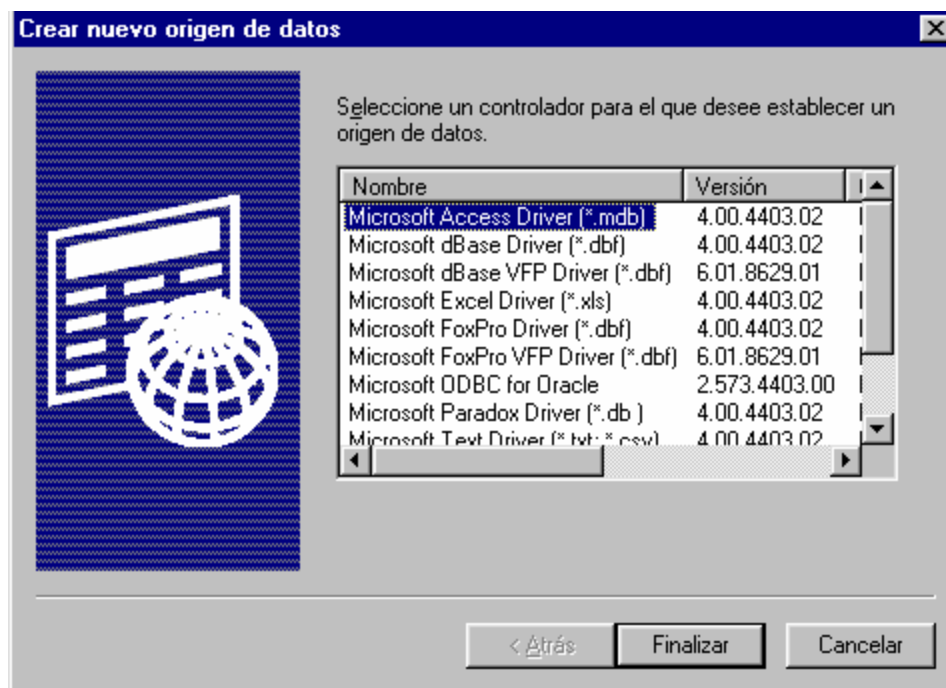


b) Seleccionar DSN de sistema.



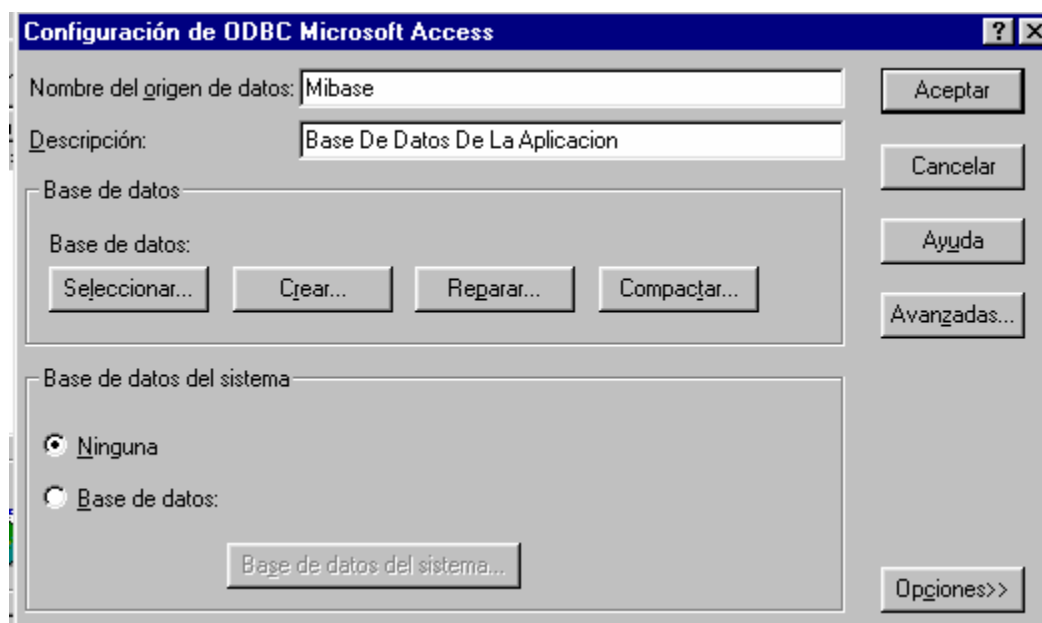


- c) Presionar Botón agregar y seleccionar el controlador requerido, luego hacer clic en finalizar para crear el origen de datos.





- d) Escribir un nombre para el origen de datos, que será el que se utiliza para referenciar a la Base De Datos física especificada por medio del botón seleccionar.



OBJETOS ADO

El modelo de Bases De Datos basado en los componentes de ActiveX Data Objects (ADO) esta formado por objetos. Estos objetos nos ofrecen una serie de métodos y propiedades que nos presentan un fácil acceso a las Bases De Batos.

Para manipular este tipo de Bases De Datos tenemos siete objetos. De los cuales tres son de gran importancia y primordiales: Connection, Recordset y Command. Los cuales se pueden instanciar directamente desde una Página Web. Completan los siete objetos: Field, Parameter, Property y Error, los cuales son utilizados como complementarios de los primordiales o principales.

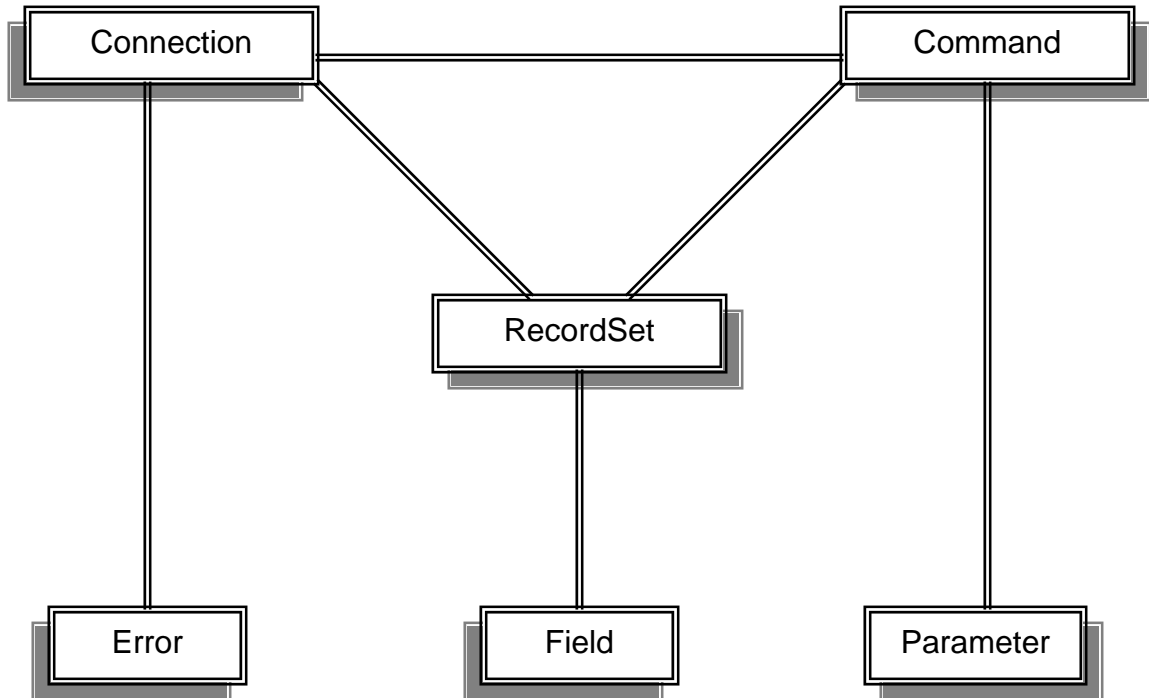


Cada objeto se define de la siguiente forma:

- **Connection:** Representa la conexión con una Base De Datos, este Objeto se utiliza para crear un enlace directo entre una página web y el servidor de Base De Datos. Mientras dure la conexión es posible realizar todas las operaciones deseadas sobre la Base De Datos. La conexión terminará cuando se indique la opción Close de este Objeto.
- **Recordset:** Representa una tabla de datos, en este Objeto será donde se almacenan las consultas realizadas a la Base De Datos a la que se esté conectado. estará formado por filas (registros) y columnas (campos) a los que se puedan acceder para exponer la información adquirida.
- **Command:** Representa un comando SQL, con este Objeto se puede ejecutar sentencias SQL sobre la Base De Datos a la que se este conectado.
- **Field:** Representa un campo de un objeto RecordSet, este Objeto solo existirá si existe el correspondiente Objeto Recordset. El Objeto Recordset lleva implícito la colección Fields que representa todos sus campos. cada elemento de esa colección es un Objeto Fields.
- **Parameter:** Representa un parámetro de un procedimiento o cuestión, es de gran ayuda al utilizar el objeto Command para lanzar procedimiento o cuestiones sobre una Base De Datos .
- **Error:** Representa un error ADO, se puede producir un error al realizar la conexión sobre la Base De Datos. esto quiere decir que este objeto solo existirá si previamente se ha intentado conectar con una Base De Datos erróneamente.
- **Property:** Representa una propiedad específica de un proveedor de datos, este objeto se encuentra un poco al margen de los demás, ya que no tiene relación alguna como ellos.



Estos Objetos descritos anteriormente poseen relaciones las cuales se describirán a continuación:



- **Connection-Recordset:** Una cualidad que tiene el objeto Connection es la de poder ejecutar comandos SQL. Con esto se puede dar el caso de que el comando devuelva un resultado, por ejemplo, si el comando realiza una consulta sobre la Base De Datos. En tal caso resultado vendrá dado dentro de un objeto Recordset.
- **Command-RecordSet:** Este es igual que el anterior, al ejecutar una consulta con el objeto Command sobre una Base De Datos el resultado viene dado en un objeto Recordset.
- **Command-Connection:** Cuando se crea un objeto Command no será utilizable hasta que no se relacione con una Base De Datos, para lo cual es necesario el objeto Connection. Con este objeto se abre una sesión con una Base De Datos y luego se asigna al objeto Command como conexión activa, y se podrán ejecutar los comandos de este objeto sobre una Base De Datos.



- **Connection- Error:** El objeto error sólo aparece cuando se produce una conexión errónea a una Base De Datos. Con este objeto y utilizando sus métodos y propiedades se pueden determinar las causas del error.
- **RecordSet-Field:** Todo objeto RecordSet tiene entre sus atributos la colección Fields. Formada por objetos del tipo Field. Estos representan cada campo de cada registro del objeto RecordSet. El objeto Field permite un acceso sencillo a la información del objeto RecordSet.
- **Command-Parameter:** Un comando SQL puede tener parámetros, los cuales pertenecerán al objeto Parameter. Además no solo se pueden ejecutar sentencias simples de SQL con el objeto Command, sino también procedimientos que probablemente llevarán parámetros, tanto de entrada como de salidas.

Conexión con una Base De datos

En primer lugar para establecer un acceso a la información de una Base De Datos hay que establecer una conexión con un origen de datos. ADO proporciona el objeto Connection utilizado para establecer y administrar las conexiones entre las aplicaciones y las Bases De Datos de ODBC.

A continuación presentamos una secuencia de comandos utilizando el objeto Connection para abrir una conexión:

```
<% 'Crear el objeto Connection par la Conexión a la Base De datos
    Set ObConn = Server.CreateObject("ADODB.Connection")
    Set ObRs = Server.CreateObject("ADODB.Recordset")
    ObRs.Open"Mibase","DSN=Estudiante;UID=Estudiante;Password=;",
        adOpenKeyset, adLockPessimistic, adCmdTable
%>
```

**Ventajas Del ASP**

Las principales ventajas que presenta la tecnología ASP son las descritas a continuación:

1. La información que existe es de naturaleza restringida. Aunque se pueden diseñar páginas Web que minimicen el ancho de banda necesario para las consultas o el procesamiento del servidor, los datos y una gran parte de los cálculos no pueden ser eliminados del lugar central.
2. ASP proporciona un adecuado soporte de procesamiento en el servidor, a la vez que permiten su funcionamiento con páginas activas en el cliente.
3. Proporciona un acceso a Bases De Datos sencillo, fiable y eficiente, apoyado por componentes ActiveX que se ejecutan únicamente en el servidor.
4. ASP devuelve al cliente páginas que se diseñan y se mantienen de forma sencilla, la cual es una característica que lo diferencia de cualquier otra tecnología existente hasta la actualidad.
5. Debido a que el código que se envía al cliente es HTML estándar correctamente la información se podrá visualizar por el cliente independientemente del navegador Web que este utilice, si posee un adecuado interprete de lenguaje.
6. Las aplicaciones realizadas con ASP no necesitan compilaciones que retardan el proceso de producción, y los errores de código no provocan la caída del servidor Web.



Desventajas del ASP

Acceder Bases De Datos desde Internet utilizando ASP es bastante sencillo, pero sin embargo este presenta algunas desventajas tales como:

1. Una desventaja bastante notable es la ejecución de páginas Web con ASP, ya que el servidor debe de ser necesariamente Windows NT/server, ya que en dicho sistema operativo es que encontramos IIS en el que ASP se encuentra ligado, por lo que se convierte en un problema a personas que trabajan con sistemas Unix, Linux, u otro sistema que no sea desarrollado por Microsoft.
2. De la anterior se derivan los problemas de licenciamiento de software para sistemas operativos requeridos y gestores de Bases De datos en instituciones o empresas que desean implementar tecnología Microsoft a las cuales se les incrementarán sus gastos monetarios.
3. Al utilizar procesamiento mixto: páginas activas en el servidor más páginas activas en el cliente, se puede presentar un error de funcionamiento en las páginas activas, debido a problemas entre los navegadores Web que utilice el cliente diferentes a Internet Explorer, como consecuencia de la no disponibilidad de un interprete de lenguaje adecuado (Ej. VBScript con Netscape) o que no sea posible cargar un control Activex.
4. Visual Basic Script es el lenguaje que normalmente utiliza ASP, este presenta un excelente manejo de Bases De Datos usando ADO, pero sin embargo en ocasiones presenta limitantes en cuanto a la validación de la información que nos suministra el cliente, habiendo que recurrir en este caso a otros lenguajes tales como JavaScript.



RESULTADO DEL ANALISIS

PLAN DEL PROYECTO

La aplicación tiene como propósito desarrollar trabajar con Base De Datos, para ello se ha seleccionado la programación de páginas ASP las cuales las utilizaremos para acceder a la Base De Datos. Las pruebas requeridas las llevaremos a cabo utilizando el Personal Web Server (PWS) y la creación de la Base de Datos en Microsoft® Access.

ESTIMACIÓN DEL PROYECTO

a) Técnicas de estimación.

Dentro de las técnicas de descomposición tenemos:

- Estimación de LDC y PF(Líneas de Código y Punto De Fusión)
- Estimación del esfuerzo.

Nuestra aplicación se descompone en subfusiones cada una de las cuales pasará por una fase de estimación individualmente una de la otra, el LDC se estima directamente y el PF indirectamente.

Las estimaciones del esfuerzo deben de descomponerse igualmente teniendo en cuenta que estas nos darán el coste de la aplicación que desarrollaremos.

Para tal actividad de estimación nos centraremos en las siguientes características:

- Complejidad de la aplicación.
- Tamaño de la aplicación.
- Grado de estructuración de la aplicación.



b) Estimaciones.

Durante la estimación de proyecto de software consideramos dos aspectos:

- ✓ Recursos.
- ✓ Esfuerzos.

Referente a los recursos tenemos:

- ✓ **Humanos:** Personal relacionado al desarrollo de la aplicación.
- ✓ **Hardware:** En el que se llevará a cabo la aplicación.
- ✓ **Software:** De soporte para el desarrollo de la misma.



RECURSOS DEL PROYECTO

a) Gente.

El equipo de trabajo del proyecto constará de 3 personas las cuales realizarán todas las etapas del proyecto.

Esta es una organización que consideramos como la más productiva.

Formada por:

- 1- Programador jefe.
- 2- Personal Técnico.
- 3- Personal de apoyo.

b) Hardware y Software.

En la parte del hardware para la realización de la aplicación utilizamos un computador que funcionará como servidor personal utilizando PWS con las siguientes características:

- Procesador Intel Pentium IV
- 128 MB de RAM.
- 20 GB de HD.
- Monitor Super VGA

En el área del Software tenemos:

- Sistema Operativo Microsoft® Windows® 98.
- Personal Web Server 4.0 para realizar las pruebas.
- Instalado DBMS Microsoft® Access 2000.
- Instalado Macro media Dreamweaver.

c) Recursos Especiales.

- Manejadores ODBC versión 4.0 de Microsoft®.



ESPECIFICACIÓN DE REQUISITOS SOFTWARE

1. Introducción

1.1 Propósito

Definición del conjunto de especificaciones de requisitos software que debe cumplir la aplicación para conectar y consultar una Base de Datos desde un servidor proxy utilizando programación ASP.

1.2 Alcance

El presente proyecto no tendrá un nombre específico debido a que su objetivo es presentar el funcionamiento del ASP (Active Server Pages) que es una herramienta eficiente para acceder a Bases De Datos desde la Web.

El producto tendrá las siguientes funciones:

- ✓ Insertar registros en la Base De Datos.
- ✓ Ver el contenido de la Base De Datos.
- ✓ Modificar registros de la Base De Datos.
- ✓ Borrar registros de la Base De Datos.
- ✓ Servicio de consultas.

1.3 Referencias.

Disponemos de una gran variedad de referencias para la realización de esta aplicación tanto en direcciones de Internet como en libros.

1.4 Visión general.

Se realizara una descripción general del producto que se desea desarrollar y luego estudiaremos cada uno de sus requisitos individualmente.



2. Descripción General.

2.1 Relaciones del producto.

La aplicación interactúa con páginas web (HTML) en conjunto con páginas ASP, las cuales se ejecutarán en los momentos que sean llamados.

2.2 Funciones del producto.

El producto Software presentado debe contener todo el código que corresponda a las diferentes funciones de la aplicación, tales como:

1. **Insertar registros en la Base de Datos**, se presentara un formulario de captura que presentará los campos a llenar para la introducción de los datos, el principal de estos será su número de carnet, y los requeridos serán nombres, apellidos, carrera, dirección domiciliar y fecha de nacimiento, sino se introduce ninguno de estos datos no podrá guardar en la Base de datos.
2. **Ver el contenido de la Base De Datos**, una vez introducido datos se podrán visualizar los datos contenidos en la Base de datos.
3. **Modificar registros de la Base De Datos**, permite la modificación de los datos que han sido ingresados, para realizar la modificación se debe de proporcionar el número de carnet y el código de la carrera.
4. **Borrar registros de la Base De Datos**, permite eliminar datos de la Base de datos proporcionando el número de carnet.
5. **Servicio de consultas**, Proporciona consultas sobre asignaturas, las cuales pueden ser accedidas a través del nombre de la carrera o por el nombre de la carrera y el año.



2.3 Características del usuario.

Por la razón de que la aplicación es únicamente para demostración del funcionamiento del ASP con respecto a la conexión de Bases De Datos con la Web, no se requiere de que los usuarios posean características particulares, solo poseer conocimientos básicos para interactuar con Internet.

2.4 Restricciones Generales.

El lenguaje de programación a utilizar será ASP en conjunto con HTML, el gestor de Bases De Datos Microsoft® Access, así como Macromedia® Dreamweaver.

3. Requisitos Específicos.

3.1 Requisitos Funcionales

3.1.1 Insertar registros en la Base De Datos.

3.1.1.1 Especificación.

3.1.1.1.1 Introducción.

Este proceso deberá realizar únicamente la inserción de los datos capturados en el formulario presentado, en sus respectivos campos de la Base de Datos, teniendo en cuenta sus validaciones.

3.1.1.1.2 Entradas.

Por pantalla:

Carnet
Nombres
Apellidos
Carrera
Dirección domiciliar
Fecha de nacimiento
Teléfono (opcional)



3.1.1.1.3 Proceso.

Se mostrará la pantalla de introducción de datos, en este proceso se hará una conexión con la Base de Datos y se almacenarán dichos datos en su tabla y campos correspondientes.

3.1.1.1.4 Salidas.

Después de la captación de datos y de su almacenamiento en la Base de Datos se mostrará una pantalla con un mensaje de que sus datos han sido insertados y se habrá un reflejo de los datos en un formulario.

3.1.1.2 Interfaces externas.

3.1.1.2.1 Interfaces de usuario.

Se presentará una página principal en la que se podrá seleccionar y hacer click en el botón insertar, se abrirá el formulario correspondiente.

3.1.1.2.2 Interfaces hardware.

Se utilizará una sola computadora, ya que utilizamos PWS para trabajar sin conexión a una red.

3.1.1.2.3 Interfaces software.

El proceso interactúa con la Base de Datos, en esta aplicación en particular utilizaremos una Base de Datos a manera de ejemplo.

3.1.1.2.4 Interfaces de comunicación.

La interfaz de comunicación es la página principal en conjunto con las páginas correspondiente a sus respectivos links.



3.1.2 Ver registros contenidos en la Base de Datos.

3.1.2.1 Especificación.

3.1.2.1.1 introducción.

Este procedimiento deberá realizar la visualización de los datos almacenados en la Base De Datos.

3.1.2.1.2 Entradas.

Por pantalla:

Bastará con seleccionar la opción correspondiente.

3.1.2.1.3 Proceso.

Se hará una conexión con la Base De Datos para luego mostrar en una tabla los registros almacenado en ella.

3.1.2.1.4 Salidas.

Se presentará por pantalla información contenida en la Base de Datos, en los campos:

Carnet
Nombres
Apellidos
Carrera

3.1.2.2 Interfaces externas.

3.1.2.2.1 Interfaces de usuario.

Se presentará una página principal en la que se podrá seleccionar y hacer click en el botón visualizar, se abrirá el formulario correspondiente.



3.1.2.2.2 Interfaces hardware.

Se utilizará una sola computadora, ya que utilizamos PWS para trabajar sin conexión a una red.

3.1.2.2.3 Interfaces software.

El proceso interactúa con la Base de Datos, en esta aplicación en particular utilizaremos una Base de Datos a manera de ejemplo.

3.1.2.2.4 Interfaces de comunicación.

La interfaz de comunicación es la página principal en conjunto con las páginas correspondiente a sus respectivos links.

3.1.3 Modificar registro de la Base de Datos.

3.1.3.1 Especificación.

3.1.3.1.1 introducción.

Este proceso restringirá el acceso para modificar los datos. Para modificar los datos hay que introducir el número de carnet y código de la carrera.

3.1.3.1.2 Entradas.

Por pantalla:

Carnet

Código de la carrera

3.1.3.1.3 Proceso.

La aplicación realiza una comparación de los datos introducidos con los datos almacenados en la Base de Datos, una vez comparados se presentará un formulario con todos los datos para realizar las modificaciones, y una vez realizada esta actividad la aplicación guardará los cambios realizados.



3.1.3.1.4 Salidas.

Sí el número de carnet y el código de la carrera son correctos el sistema enviará todos los datos correspondientes por pantalla, una vez modificado los datos y sean enviados, se enviará un mensaje de notificación que los datos han sido actualizados.

Si los datos introducidos no son correctos el sistema enviará un mensaje notificando que esta fallando ya sea el número de carnet o el código de la carrera.

3.1.3.2 Interfaces externas.

3.1.3.2.1 Interfaces de usuario.

Se presentará una página principal en la que se podrá seleccionar y hacer click en el botón modificar, se abrirá el formulario correspondiente.

3.1.3.2.2 Interfaces hardware.

Se utilizará una sola computadora, ya que utilizamos PWS para trabajar sin conexión a una red.

3.1.3.2.3 Interfaces software.

El proceso interactúa con la Base de Datos, en esta aplicación en particular utilizaremos una Base de Datos a manera de ejemplo.

3.1.3.2.4 Interfaces de comunicación.

La interfaz de comunicación es la página principal en conjunto con las páginas correspondiente a sus respectivos links.



3.1.4 Borrar registro de la Base de Datos.

3.1.4.1 Especificación.

3.1.4.1.1 introducción.

Este proceso restringirá el acceso para borrar los datos. Para borrar datos se pedirá únicamente el número de carnet.

Este proceso es poco común en una aplicación WEB por seguridad e integridad de los datos los cuales se encuentran almacenados en el respectivo servidor, presentamos la manera en que se realiza este proceso ya que uno de nuestros objetivos es realizar un estudio de ASP.

3.1.4.1.2 Entradas.

Por pantalla:

Carnet.

3.1.4.1.3 Proceso.

La aplicación realiza una comparación de el número de carnet introducido con los datos almacenados en la Base de Datos, una vez comparados se procede con el borrado de los datos.

3.1.4.1.4 Salidas.

Sí el número de carnet es correcto el sistema procederá con el borrado de los datos correspondientes, una vez borrado los datos, se enviará un mensaje de notificación que los datos han sido borrados.

Si los datos introducidos no son correctos el sistema enviará un mensaje notificando que el número de carnet no existe en la Base de Datos.



3.1.4.2 Interfaces externas.

3.1.4.2.1 Interfaces de usuario.

Se presentará una página principal en la que se podrá seleccionar y hacer click en el botón borrar, se abrirá el formulario correspondiente.

3.1.4.2.2 Interfaces hardware.

Se utilizará una sola computadora, ya que utilizamos PWS para trabajar sin conexión a una red.

3.1.4.2.3 Interfaces software.

El proceso interactúa con la Base de Datos, en esta aplicación en particular utilizaremos una Base de Datos a manera de ejemplo.

3.1.4.2.4 Interfaces de comunicación.

La interfaz de comunicación es la página principal en conjunto con las páginas correspondiente a sus respectivos links.

3.1.5 Consultas con ASP.

3.1.5.1 Especificación.

3.1.5.1.1 introducción.

En este procedimiento se llevan a cabo dos consultas, en la primera se debe de introducir el nombre de la carrera y en la segunda el nombre de la carrera y el año de estudio correspondiente, cada una de las consultas funcionan independiente una de la otra.



3.1.5.1.2 Entradas.

Por pantalla:

Consulta 1:

Nombre de la carrera

Consulta 2:

Nombre de la carrera

Año de estudio

3.1.5.1.3 Proceso.

El sistema hará una comparación de los datos introducidos con los datos ya establecidos en la codificación del sistema.

3.1.5.1.4 Salidas.

Consulta 1: Sí el nombre de la carrera es correcta se presentará un formulario con los datos de la carrera correspondiente.

Consulta 2: Sí el nombre de la carrera y el año de estudio son correctos se presentará un formulario con los datos de la carrera correspondiente al año enviado.

Si el nombre de la carrera en la consulta 1, y el nombre de la carrera y el año en la consulta 2 son incorrectos se enviará un mensaje de acceso denegado.

3.1.5.2 Interfaces externas.

3.1.4.2.1 Interfaces de usuario.

Se presentará una página principal en la que se podrá seleccionar y hacer click en el botón consultas, se abrirá el formulario correspondiente.



3.1.4.2.2 Interfaces hardware.

Se utilizará una sola computadora, ya que utilizamos PWS para trabajar sin conexión a una red.

3.1.4.2.3 Interfaces software.

El proceso interactúa con la Base de Datos, en esta aplicación en particular utilizaremos una Base de Datos a manera de ejemplo.

3.1.4.2.4 Interfaces de comunicación.

La interfaz de comunicación es la página principal en conjunto con las páginas correspondiente a sus respectivos links.

3.2 Requisitos de Funcionamiento.

- Requisitos estáticos: Personal Web Server 4.0, para las pruebas y funcionalidad de la aplicación.
- Requisitos Dinámicos: Establecer una conexión entre las páginas ASP y la Base De Datos.

3.3 Restricciones de Diseño.

El formato de la aplicación no contiene datos específicos, más bien son particulares y se trabajan solo a manera de ejemplo.

3.4 Atributos.

3.4.1 Seguridad

Este es un aspecto importante ya que nuestra aplicación final interactúa con páginas ASP, las cuales deben de ejecutarse correctamente para el buen funcionamiento en conjunto de toda la aplicación.



3.4.2 Mantenimiento.

Cualquier modificación que afecte a los requisitos mencionados en este documento debe de ser reflejada en el mismo, así como la documentación de las fases de análisis y diseño.

3.5. Otros Requisitos.

3.5.1. Base de Datos.

El almacenamiento de la información se realizará por medio del uso de una Base de Datos relacional.

3.5.2 Entorno de programación ASP.

La programación de la aplicación se realizará con la tecnología ASP ya que es el que reúne los componentes necesarios para crear aplicaciones para Internet que se ejecutan en un servidor, y en nuestro caso que es un servidor proxy.

3.5.3 Operaciones.

Todas las operaciones de la aplicación se realizan sobre la Base de Datos, tales como: almacenamiento de la información y evaluación de campos.



DISEÑO DE DATOS

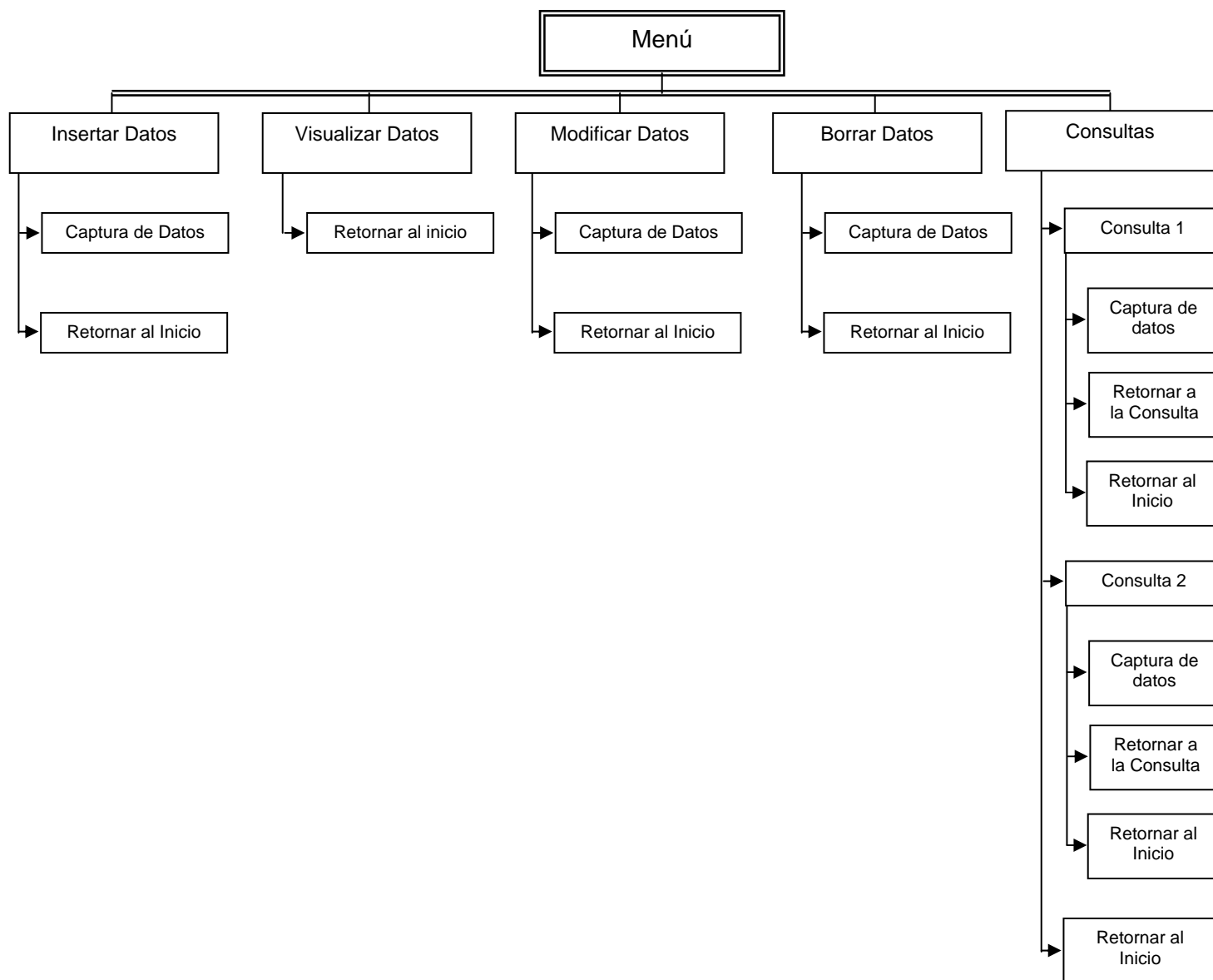
El diseño de datos nos presenta la estructuración de la Base de Datos con la que interactúa la aplicación, la cual tendrá las siguientes tablas:

estudiante : Tabla			
	Nombre del campo	Tipo de datos	Descripción
	Id	Autonumérico	Llave primaria de la tabla
	codigo	Texto	Código generado por medio de código
	carnet	Texto	Numero de carnet del estudiante
	nombres	Texto	Nombre del estudiante
	apellidos	Texto	Apellidos del estudiante
	carrera	Texto	Carrera que cursa el estudiante
	direccion	Texto	Dirección domiciliar
	fechanac	Texto	Fecha de nacimiento
	telefono	Texto	Telefono
	dia	Texto	Día de su nacimiento
	mes	Texto	Mes de nacimiento
	ano	Texto	Año de nacimiento

carrera : Tabla			
	Nombre del campo	Tipo de datos	Descripción
	codcarr	Texto	Código de la carrera
	carrera	Texto	Nombre de la carrera
	codasig	Texto	Código de la asignatura
	nombreasig	Texto	Nombre de la asignatura
	ano	Texto	Año de la carrera
	semestre	Texto	Semestre al que pertenece la carrera



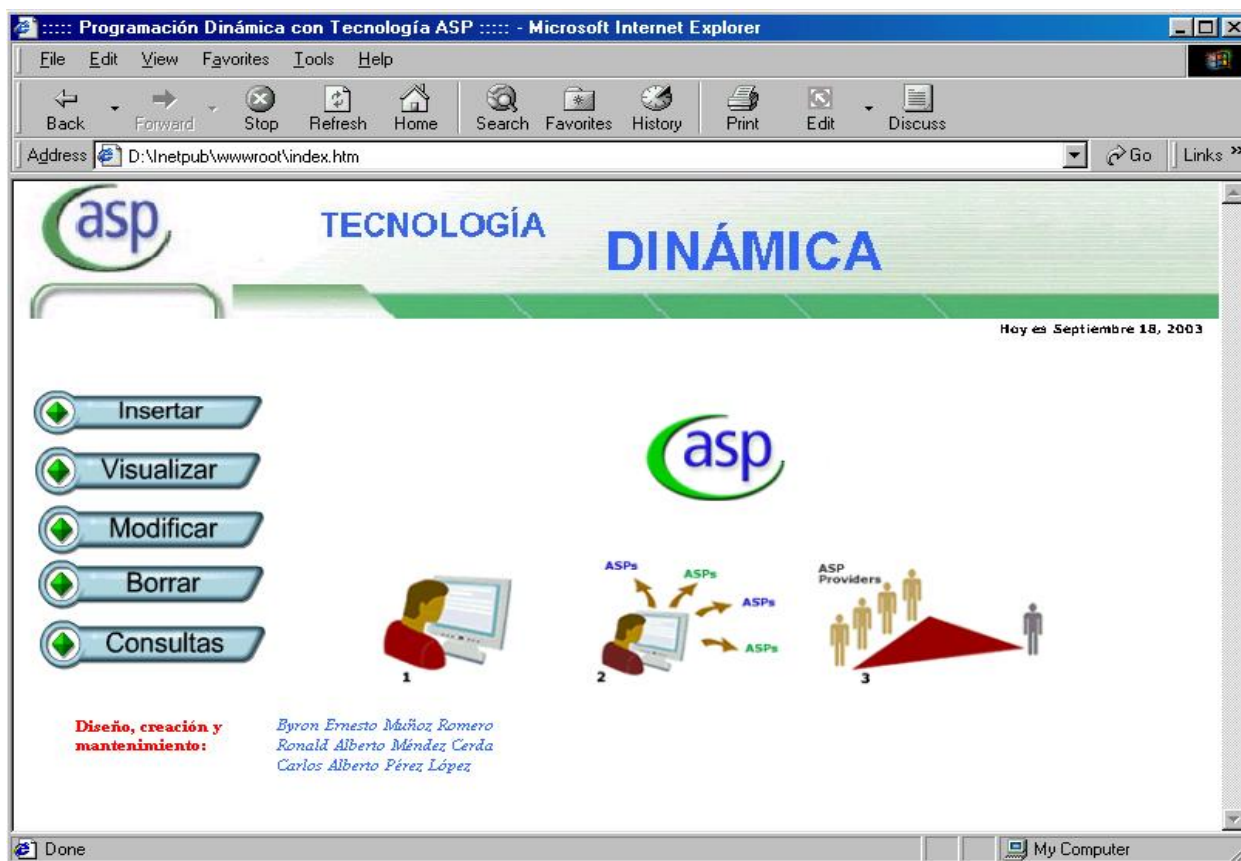
DISEÑO ARQUITECTÓNICO





DISEÑO DE INTERFAZ Y CODIFICACION

Este es el menú principal de la aplicación donde se visualizan cada una de las opciones y se accede a cada una de ellas con un simple clic.





Conexión de Base de Datos con la WEB utilizando ASP



UNAN - León

UNAN -León

Se logra ver que aparecen las opciones siguientes:



Al hacer un clic sobre el botón Insertar se ingresa a un formulario donde se podrán hacer la inserción de datos.

The screenshot shows a Microsoft Internet Explorer window with the title 'Página De Insercion De Datos - Microsoft Internet Explorer'. The address bar shows 'C:\My Documents\wwwroot\insertar.htm'. The page content includes a header with 'asp' logo and 'TECNOLOGÍA DINÁMICA'. The form fields are as follows:

Field Label	Value
Carnet	97-00291-1
Nombres	BYRON ERNESTO
Apellidos	MUÑOZ ROMERO
Carrera	COMPUTACION
Dirección Domiciliar	CHINANDEGA, NICARAGUA
Fecha De Nacimiento	08 / 03 / 1983
Teléfono	0341-3481

At the bottom of the form, there are two buttons: 'Insertar Datos' and 'Limpiar Datos'. A link labeled 'Página Principal' with a left arrow is also present.

Este proceso se verificará con la aparición de otro formulario que reflejara algunos de los datos recién insertados, con opción de regresar a la página principal.



Conexión de Base de Datos con la WEB utilizando ASP



UNAN - León

UNAN -León

Este código permite el almacenamiento de los datos en los correspondientes campos de la Base de Datos.

1. 'Tomamos los valores de los campos del formulario y lo guardamos en las variables

```
<!--#INCLUDE File="Adovbs.inc"-->  
  
<%  
    strcarnet = cStr(Request.Form("carnet"))  
    strNombres = cStr(Request.Form("Nombres"))  
    strApellidos = cStr(Request.Form("Apellidos"))  
    strcarrera = cStr(Request.Form("carrera"))  
    strDireccion = cStr(Request.Form("Direccion"))  
    strFechadeNac = cStr(Request.Form("fechanac"))  
    strtelefono = cStr(Request.Form("telefono"))  
    strdia = cStr(Request.Form("dia"))  
    strmes = cStr(Request.Form("mes"))  
    strano = cStr(Request.Form("ano"))
```

2. 'Creamos la conexión a la base de datos

```
Set ObConn = Server.CreateObject("ADODB.Connection")  
Set oRs = Server.CreateObject("ADODB.Recordset") ObConn.Open  
"datos"  
oRs.Open"Estudiante",ObConn,adOpenKeySet,adLockPessimistic,  
adCmdTable  
oRS.Filter = "Carnet=" & Request.Form("Carnet") & " "  
  
IF ORS.EOF THEN
```



3. 'Damos la instrucción de que se agregue un registro nuevo a la Base de Datos

```
oRs.AddNew
```

4. 'Llenamos los campos de la Base de Datos con los valores de las variables

```
oRs.Fields("Carnet") = strcarnet  
oRs.Fields("Nombres") = strNombres  
oRs.Fields("Apellidos") = strApellidos  
oRs.Fields("Carrera") = strcarrera  
oRs.Fields("Direccion") = strDireccion  
oRS.Fields("Fechanac") = strFechanac  
oRs.Fields("Telefono") = strTelefono  
oRs.Fields("dia") = strdia  
oRs.Fields("mes") = strmes  
oRs.Fields("ano") = strano
```

5. 'Guardamos los cambios

```
oRs.Update %>
```




Esta opción permite el ingreso a un formulario en el cual se reflejan los datos contenidos en la Base de Datos de la aplicación.

No. CARNET.	NOMBRES	APELLIDOS	CARRERA
97-00291-1	BRYON ERNESTO	MUÑOZ ROMERO	COMPUTACION
281-0281-2	RONALD ALBERTO	MENDEZ CERDA	COMPUTACION
281-0281-3	CARLOS ALBERTO	PEREZ LOPEZ	COMPUTACION

Observamos que se muestran registros de la Base de Datos ordenados en una tabla, y se tiene un enlace que regresa a la página principal.



Conexión de Base de Datos con la WEB utilizando ASP



UNAN - León

UNAN -León

El código siguiente hace que los datos almacenados en la Base de Datos se reflejen en un formulario contenido en una página Web.

```
<!-- #include FILE="adovbs.inc" -->
<%
set Ob_Conn=Server.CreateObject("ADODB.Connection")
Set Ob_RS=Server.CreateObject("ADODB.RecordSet")
Ob_Conn.Open "Datos"
Set Ob_RS= Ob_Conn.Execute("Select * from estudiante")%>
</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<TABLE BORDER=1>
  <TR>
    <TH width="131"><b><font color="#FF0000" face="Arial, Helvetica, sans-serif">No. CARNET.</font></b></TH>
    <TH width="301"><b><font color="#FF0000" face="Arial, Helvetica, sans-serif">NOMBRES</font></b></TH>
    <TH width="248"><b><font color="#FF0000" face="Arial, Helvetica, sans-serif">APELLIDOS</font></b></TH>
    <TH width="159"><b><font color="#FF0000" face="Arial, Helvetica, sans-serif">CARRERA</font></b></TH>
    <% DO UNTIL Ob_RS.EOF %>
  <TR>
    <TD width="131"><i><font color="#0000FF"><%=Ob_RS("carnet")%></font></i></TD>
    <TD width="301"><i><font color="#0000FF"><%=Ob_RS("NOMBRES")%></font></i></TD>
    <TD width="248"><i><font color="#0000FF"><%=Ob_RS("APELLIDOS")%></font></i></TD>
    <TD width="159"><i><font color="#0000FF"><%=Ob_RS("carrera")%></font></i></TD>
    <% Ob_RS.MoveNext
  LOOP
  Ob_RS.Close
  Ob_Conn.Close%>
</TABLE>
```



En el botón Modificar al darle un simple clic permite ingresar a una página que contiene un formulario donde se capturan los datos correspondientes al registro contenido en la Base de Datos que se desea modificar.

Esta página pide número de carnet y código de la carrera y si estos son verdaderos y concuerdan se presentará una nueva pantalla con los datos correspondientes y desde la cual podrán ser modificados.

Si los datos son verdaderos estos deben de ser comprobados por una función la cual se presentará a continuación:



1. Creamos la conexión con la Base de Datos:

```
<!-- #include FILE="adovbs.inc" -->  
<%  
dim ob_Conn  
dim oRS  
  
if Request.Form("hModificar")="Modificar" then  
set Ob_Conn=Server.CreateObject("ADODB.Connection")  
Set oRs = Server.CreateObject("ADODB.Recordset")  
Ob_Conn.Open "Datos"
```

2. Ejecutamos la Consulta:

```
CONSULTA = "SELECT * FROM estudiante WHERE Carnet=" &  
Request.Form("Carnet") & "" & " " & "or" &  
" " & "codigo=" & request.form("codigo") & ";"  
Set oRS = Ob_Conn.Execute(CONSULTA)
```



3. Comprobamos que los datos introducidos sean correctos:

```
IF not oRS.EOF THEN
    if ((oRs("carnet") <> request.form("carnet")) and (oRS("codigo")=
request.form("codigo"))) then%>
        <SCRIPT LANGUAGE = "VBSCRIPT">
            MsgBox "Revisa Tu Número De Carnet"
            window.history.back()
        </SCRIPT>
    <%end if%>
    <%if ((oRs("codigo") <> request.form("codigo")) and (oRS("carnet")=
request.form("carnet"))) then%>
        <SCRIPT LANGUAGE="VBSCRIPT">
            MsgBox "El Código No Coincide Con El Numero De Carnet"
            window.history.back()
        </SCRIPT>
    <%end if%>
```

4. Presentamos los datos contenidos en la Base de Datos:

```
<%
strcarnet=oRS.Fields("carnet")
strnombres = oRS.Fields("nombres")
strapellidos = oRS.Fields("apellidos")
strcarrera = oRS.Fields("carrera")
strdia = oRS.Fields("dia")
strmes = oRS.Fields("mes")
strano = oRS.Fields("ano")
strdireccion = oRS.Fields("direccion")
strtelefono = oRS.Fields("telefono") %>
```



5. Realizamos la actualización de los cambios:

```
<%else
if(not (Ob_RS.EOF) AND (Request.Form("ncarnet") <>
Request.Form("carnet"))))then

    conn.execute "update Estudiante Set carnet='" & request.form("carnet") & "'"
    & " where Carnet='" & request.form("Carnet") & "';"
    conn.execute "update Estudiante Set nombres='" & request.form("nombres")
    & "'" & " where Carnet='" & request.form("Carnet") & "';"
    conn.execute "update Estudiante Set apellidos='" & request.form("apellidos")
    & "'" & " where Carnet='" & request.form("Carnet") & "';"
    conn.execute "update Estudiante Set carrera='" & request.form("carrera")&
    "'" & " where Carnet='" & request.form("Carnet") & "';"
    conn.execute "update Estudiante Set direccion='" &
    request.form("direccion") & "'" & " where Carnet='" & request.form("Carnet") &
    "';"
    'conn.execute "update Estudiante Set fechanac='" &
    request.form("fechanac") & "'" & " where Carnet='" & request.form("Carnet") &
    "';"
    conn.execute "update Estudiante Set telefono='" & request.form("telefono")
    & "'" & " where Carnet='" & request.form("Carnet") & "';"
    conn.execute "update Estudiante Set dia='" & request.form("dia") & "'" & "
    where Carnet='" & request.form("Carnet") & "';"
    conn.execute "update Estudiante Set mes='" & request.form("mes") & "'" & "
    where Carnet='" & request.form("Carnet") & "';"
    conn.execute "update Estudiante Set ano='" & request.form("ano") & "'" & "
    where Carnet='" & request.form("Carnet") & "';"
%>
```



6. Cerramos la conexión con la Base de Datos:

```
oRS.close  
set oRS = nothing  
Ob_Conn.Close  
set Ob_Conn = nothing
```



Permite acceder a una página donde se muestra un formulario de captura de datos, para especificar el registro que se desea borrar de la Base de Datos.



Esta página pide número de carnet si este existe se borra el registro correspondiente y se presentará una nueva pantalla indicando que el registro ha sido borrado. Si el número de carnet no existe aparecerá un mensaje que el carnet no existe en la Base de Datos.



Conexión de Base de Datos con la WEB utilizando ASP



UNAN - León

UNAN -León

Esta parte de borrado no es muy común ya que puede darse el caso de la pérdida de información no deseada. A continuación se presenta parte del código para efectuar el borrado de un registro existente en la Base de Datos.

1. Tomamos los valores de los campos del formulario y lo guardamos en las variables

```
<!--#INCLUDE File="Adovbs.inc"-->  
<%  
    strcarnet = cStr(Request.Form("carnet"))
```

2. Creamos la conexión a la base de datos.

```
Set ObConn = Server.CreateObject("ADODB.Connection")  
Set oRs = Server.CreateObject("ADODB.Recordset")  
ObConn.Open "datos"  
ORs.Open "Estudiante",ObConn, adOpenKeyset, adLockPessimistic,  
adCmdTable
```

3. Filtramos por el número de carnet.

```
ors.filter="Carnet=" & request.form("carnet")&" "  
if ors.EOF then %>  
<script language="Javascript">  
    alert("Este número de carnet no existe")  
    window.history.back()  
</script>
```




4. Damos la orden de que se borre el registro y se actualicen los cambios .

```
<% else
    oRs.delete
    oRs.update %>
<CENTER>
<%response.write"    EL    REGISTRO    HA    SIDO    ELIMINADO
SATISFACTORIAMENTE, regrese al inicio... " %>
<BR>
<BR>
<% end if    %>
<A HREF="principal.htm"> <B> </B></A><b> </b>
```

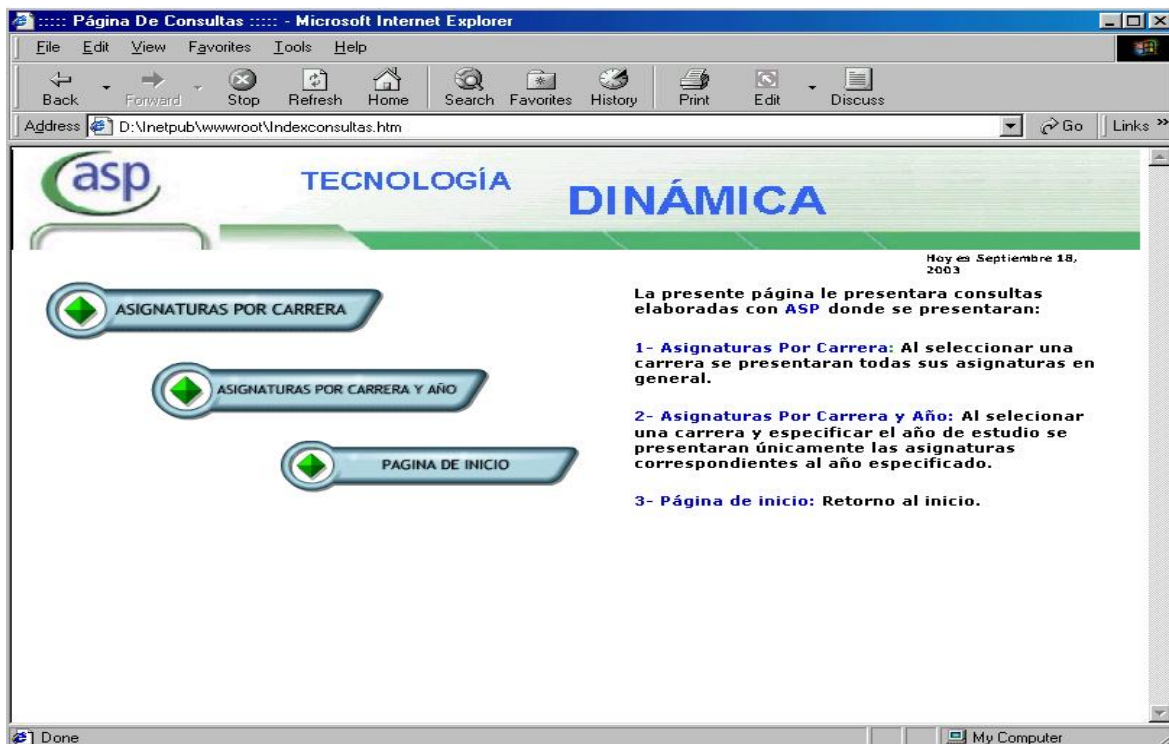
5. Cerramos la conexión y liberamos los recursos para ser utilizados de nuevo.

```
<%Cerramos la conexion
oRs.Close
ObConn.Close

'Liberamos los recursos para que se pueda volver a usar inmediatamente
'Set oRs = Nothing
'Set ObConn = Nothing
%>
```



Brinda consultas elaboradas con ASP, al hacer clic en el botón Consultas se presentará una página donde habrán dos botones para dos tipos de consultas, con acceso independiente cada una y presentando una página por consulta que contiene un formulario de captura de datos.



Como se puede apreciar esta es la página para acceder a las consultas, se pueden consultar Asignaturas por carrera, Asignaturas por carrera y año y hay opción también de regresar a la página de inicio o principal.



Presentará una página conteniendo un formulario y en donde se pide el nombre de la carrera la cual estará contenida en un combo.

The screenshot shows a Microsoft Internet Explorer window titled "Página De Consultas :::: - Microsoft Internet Explorer". The address bar shows "http://desarrollomar/consulta.htm". The page has a header with "asp" logo and "TECNOLOGÍA DINÁMICA". The main content area is titled "PANTALLA PARA CONSULTAR SUS DATOS" and contains the following text:

Esta consulta le presentara todas las asignaturas de la carrera que usted seleccione, realice lo siguiente:

1. Seleccione el **Nombre de la Carrera** que usted desee conocer sus asignaturas.

Below the text is a form with a green header "CARRERA". Inside the form, there is a label "Codigo de la carrera:" followed by a dropdown menu showing "COMPUTACION". Below the form, there is an example: "(Ejemplo: Nombre de la Carrera: Computación)". At the bottom of the form, there are two buttons: "Enviar Datos" and "Limpiar Datos". To the left of the buttons is a green arrow pointing left with the text "Atrás". To the right is a house icon with the text "Página Principal". The status bar at the bottom shows "Listo" and "Intranet local".

Luego si la carrera existe se presentarán en una nueva página las asignaturas correspondientes a la carrera seleccionada.



Conexión de Base de Datos con la WEB utilizando ASP



UNAN - León

UNAN -León

The screenshot shows a Microsoft Internet Explorer window titled "Página De Consultas - Microsoft Internet Explorer". The address bar displays "http://desarrollomar/consultaservtec.asp". The page features a header with the "asp" logo and the text "TECNOLOGÍA DINÁMICA". Below the header, the text "REGISTROS ENCONTRADOS" is displayed in red. There are two navigation links: "Atrás" with a left arrow and "Página de Consultas" with a right arrow. A table with four columns is shown: "CODIGO CARRERA", "NOMBRE CARRERA", "CODIGO ASIGNATURA", and "ASIGNATURA". The table contains three rows of data.

CODIGO CARRERA	NOMBRE CARRERA	CODIGO ASIGNATURA	ASIGNATURA
001	COMPUTACION	0001	CALCULO I
001	COMPUTACION	0002	PROGRAMACION I
001	COMPUTACION	0003	BASE DE DATOS

Hay opción de regresar a la página anterior de captura de datos y también para ir a la página de consultas.



Presentará una página conteniendo un formulario y en donde se pide el nombre de la carrera la cual estará contenida en un combo, y el año de estudio.

The screenshot shows a Microsoft Internet Explorer window with the address bar displaying `http://desarrollomar/consultaUNO.htm`. The page title is "Página de Consultas ::: - Microsoft Internet Explorer". The main content area has a green header bar and the title "PANTALLA PARA CONSULTAR SUS DATOS" in red. Below the title, there is a paragraph: "Esta consulta le presentara todas las asignaturas de la carrera y del año que usted seleccione, realice lo siguiente:". This is followed by two numbered instructions: "1. Seleccione el **Nombre de la Carrera** que usted desee conocer sus asignaturas." and "2. Ingrese el **Año de la Carrera**". Below these instructions is a form titled "CARRERA" with a teal header. The form contains two fields: "Nombre de la carrera:" with a dropdown menu showing "COMPUTACION" and "Año de la carrera" with a text input field containing the number "5". Below the form, there is an example text: "(Ejemplo: Nombre de la Carrera: **Computación**, Año de la carrera: **5**)". At the bottom of the form, there are three buttons: "Atrás" (with a green arrow icon), "Enviar Datos", and "Limpiar Datos". To the right of these buttons is a "Página Principal" link with a home icon. The status bar at the bottom of the browser shows "Intranet local".

Luego si la carrera existe se presentarán en una nueva página las asignaturas correspondientes a la carrera seleccionada.



The screenshot shows a Microsoft Internet Explorer window titled "Página de Consultas :::: - Microsoft Internet Explorer". The address bar displays "http://desarrollomar/consultaUNO.asp". The page features a header with the "asp" logo and the text "TECNOLOGÍA DINÁMICA". Below the header, there are navigation links: "Atrás" (with a left arrow) and "Página de Consultas" (with a right arrow). The main content area displays a table titled "REGISTROS ENCONTRADOS". The table has six columns: "CODIGO CARRERA", "NOMBRE CARRERA", "CODIGO ASIGNATURA", "ASIGNATURA", "AÑO", and "SEMESTRE". The first row of data shows: "001", "COMPUTACION", "0003", "BASEDEDATOS", "2", and "1". A large, semi-transparent "asp" watermark is visible in the background of the page content.

CODIGO CARRERA	NOMBRE CARRERA	CODIGO ASIGNATURA	ASIGNATURA	AÑO	SEMESTRE
001	COMPUTACION	0003	BASEDEDATOS	2	1

Hay opción de regresar a la página anterior de captura de datos y también para ir a la página de consultas.



CONCLUSIÓN

Con la elaboración de este trabajo se ha presentado el funcionamiento de la tecnología ASP, en especial trabajando con base de datos desde la Web, permitiendo la interacción dinámica con las bases de datos.

La programación en ASP es fácil de manejar, eficiente, confiable y con un alto grado de seguridad en la manipulación de datos, lo que nos indica que cualquier aplicación WEB en la que se use programación ASP es un sitio seguro.

Se pueden desarrollar sitios WEB donde se lleven las notas de los alumnos de una determinada Universidad o colegio, sistemas de consultas de estado de cuenta de tarjetas de crédito, de cuentas bancarias, venta en línea, etc.

Cabe señalar que este trabajo fue elaborado como un modelo que muestra la conexión de base de datos con sitios Web utilizando la tecnología ASP para esto elaboramos un estudio de esta tecnología y utilizamos un servidor personal para la prueba de las páginas.

Se hizo uso de un servidor local con Windows 98, el PWS (Personal Web Server 4.0) y la programación que se implementó fue Visual Basic Script y java Script.



BIBLIOGRAFÍA

- Bobadilla Sancho Jesús, Alcocer Jarabe Alejandro. Creación de aplicaciones WEB en Windows NT Active Server Page.

OTRAS FUENTES DE INFORMACIÓN

- <http://www.asptutor.com>: Sitio sobre programación en ASP
- <http://www.asp.com>: Sitio sobre programación en ASP
- [http://www. Geonet.com/web/guillermo/asp](http://www.Geonet.com/web/guillermo/asp): Sitio Web con ejemplos concretos de programación en ASP
- <http://www.aspfacil.com>: Sitio web para encontrar ejemplos básicos de programación en ASP.
- <http://www.lawebdelprogramador.com>: Sitio web con ejemplos y recomendaciones de programación.
- <http://www.monografias.com>: Sitio para encontrar manuales de programación



ANEXOS

LOS OBJETOS COMPONENTES DE ADO

Propiedades y métodos mas relevantes del objeto connection.

ConnectionString

Es una cadena de caracteres con la información necesaria para establecer una conexión con la fuente de datos. Por tanto, es la propiedad básica de este objeto. Aunque hay hasta siete argumentos distintos que se pueden suministrar en esta cadena, los básicos son el DSN que identifica al archivo de Base de Datos y el login y password si existen. Los argumentos se separan con punto y coma.

Ejemplo: "DSN = Mydsn; UID = Mylogin; PWD = Mycontraseña"

Open

Abre la conexión con la Base de Datos. Si antes hemos asignado la propiedad connectionString, este método no necesita parámetros.

Close

Cierra la conexión con la Base de Datos.

Propiedades y métodos mas importantes del Recordset.

Para una mejor comprensión y puesto que son numerosos, los hemos dividido por categorías de utilidad.

Propiedades que hacen referencia al origen de los datos:

Estas dos propiedades deben ser asignadas a todo Recordset, pues le dicen la Base de Datos y la tabla de la que se obtienen sus datos.

**ActiveConnection**

Como se ha comentado antes de esta propiedad se le debe asignar un objeto *Connection* que se halla creado previamente. Indicará al *Recordset* la Base de Datos en la que se buscará la tabla.

Source

Indica al objeto *Recordset* la tabla a la que representará. A la propiedad *Source* se le asigna normalmente una cadena de caracteres con el nombre de la tabla. Sin embargo, también es posible asignarle una sentencia SQL y entonces el objeto *Recordset* referenciará al resultado de aplicar dicha sentencia.

LockType

Indica el tipo de bloqueo que se realiza sobre la Base de Datos. Por defecto toma el valor **adLockReadOnly** que como su nombre indica solo sirve para leer datos. Si deseamos editar o añadir registros, tendremos que cambiar esta propiedad por otro valor. El más usado es **adLockOptimistic** que permite editar la Base de Datos o añadir registros realizando un bloqueo optimista (sólo cuando sea estrictamente necesario)

CursorType

El tipo de cursor que se utiliza para recorrer los registros de un *Recordset*. Por defecto toma el valor **adOpenForwardOnly** que como su nombre indica solo permite moverse hacia adelante. Por ello si queremos utilizar libremente todos los métodos de movimientos de un *Recordset* (*MoveFirst*, *MoveNext*, *MoveTo*, *MovePrevious*, etc) tendremos que cambiar el cursor por uno más potente. Esto puede hacerse asignando a esta propiedad el valor **adOpenKeySet**.

**Propiedades que hacen referencia al número de registros:****RecordCount**

Número de registros de la tabla a la que representa el objeto Recordset.

Ejemplo: <H3> Tenemos <%= rstEstudiantes.RecordCount%> Estudiantes registrados en nuestra Base de Datos </H3>

Eof

Acrónimo de End of File. Vale TRUE si estamos después del último registro y FALSE si no. Se usa mucho como condición en Bucles While, los cuales se ejecutan hasta llegar al último registro.

Bof

Acrónimo de Bejín Of File. Vale TRUE si estamos antes del primer registro y FALSE si no.

Métodos para mover el cursor (registro activo):

Estos métodos pueden funcionar todos o no dependiendo del tipo de cursor creado. El tipo de cursor se asigna en la propiedad CursorType. Por ejemplo, para asignar un cursor adOpenKeySet que permita moverse hacia delante o hacia atrás:

MoveFirst

Mueve el cursor al primer registro de la tabla.

MoveLast

Mueve el cursor al último registro de la tabla.-

MoveNext

Mueve el cursor al siguiente registro.

**MovePrevious**

Mueve el cursor al registro anterior.

Lectura y modificación de los campos del registro activo

La sintaxis para acceder a los datos de un campo del registro activo en un Recordset es:

MiRecordset ("Domicilio")

Leemos el valor del campo domicilio del registro activo y se lo asignamos a la variable dom. Con la expresión.

```
MiRecordset("Domicilio") = "C/ Bretón de los Herreros 19, 1º M" miRecordset.Update.
```

Asignamos un valor al campo Domicilio del registro activo. Tras la edición del registro, es necesario llamar al método Update. El motivo es que los cambios en el registro activo se realizan sobre un buffer(espacio de almacenamiento intermedio) y no sobre el registro propiamente dicho.

Método para agregar o eliminar registros de la tabla.**Delete**

Elimina el registro activo es muy fácil. Basta con invocar este método.

AddNew y Update

Crear un nuevo registro involucra dos métodos: Primero AddNew crea un nuevo registro en blanco. Después asignamos valores a los distintos campos del registro. Por último invocamos el métodos Update para que se haga efectiva la incorporación del nuevo registro con los valores asignados.

Recuerde que para que esto funcione hay que asignar un tipo de bloqueo que permita la edición de la Base de Datos. Esto se hace asignando a la propiedad LockType del Recordset uno de estos valores: adLockOptimistic, adLockPessimistic o adLockBatchOptimistic.

**Propiedades y métodos más relevantes del objeto command****ActiveConnection**

Es una referencia al objeto connection que enlaza con la Base de Datos. Es imprescindible asignar esta propiedad antes de invocar el método command.execute para ejecutar el comando.

CommandText

Este es el texto del comando. Si se trata de una consulta SQL. (lo habitual), esta propiedad es simplemente una cadena con el texto de la consulta.

Execute

El método que ejecuta la consulta. El resultado de la ejecución del comando normalmente será un Recordset.