



Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

Ingeniería en Electrónica y Telecomunicaciones

“Sistema de Administración
de Red (S.A.R.) Versión 1.0”

T E S I S

Que para obtener el título de Ingeniero en Electrónica y
Telecomunicaciones

Presenta

Ávila Flores José Salvador

Asesor: Lic. Arturo Austria Cornejo

Pachuca de Soto Hidalgo, Noviembre 2005



Agradecimientos

El presente trabajo esta dedicado primeramente a *DIOS*, por darme la oportunidad de llegar a esta instancia.

A mis *Padres* por todo el apoyo que me han brindado en todo momento.

A mis *Hermanos* Pablo Andrés y Mariana por estar siempre conmigo.

A *Verónica* por todo el apoyo que me brindaste durante mi carrera.

A mis *Maestros* por dar su máximo esfuerzo y enseñarnos lo mejor para que nos desarrolláramos profesionalmente.

A todos mis *Compañeros* que formaron parte de esta primera generación de *Electrónicos* de los cuales aprendí muchas cosas.

De una manera muy especial al *Lic. Arturo Austria* por hacer de este trabajo una realidad.

Al *In.g. Miguel A. Rosas Yacotu*, coordinador de nuestra carrera por dar lo mejor y brindarnos todo el apoyo durante nuestra formación profesional.

A *todas las personas* que contribuyeron de la mejor manera para alcanzar el éxito y que no aparecen porque sería una lista interminable a todos *GRACIAS*.

Dedicatorias

A DIOS por estar siempre a mi lado y no dejarme caer en ningún momento.
Gracias.

A mi *Padre, Andrés Ávila Aguilar*, por darme todo ese apoyo y darme este legado que es mi carrera profesional.

A mi *Madre, Ana María Flores Cano*, gracias por los cuidados y las atenciones que solo una madre nos puede brindar.

A mi *Hermano Pablo Andrés*, gracias por estar siempre en los momentos llenos de incertidumbre y ayudarme a salir adelante.

A mi *Hermana Mariana*, por estar siempre presente y brindarme todo el apoyo y darme consejos que ayudaron a salir adelante.

A ti *hijo mío*, por ti estoy haciendo mi mayor esfuerzo, para que tengas lo mejor del mundo y cuando puedas leer estas líneas sepas de todo esto fue realizado por ti.

A mi *Querida amiga Mariana Mendoza Cid*, por ser una persona tan maravillosa conmigo.

CONTENIDO

Agradecimientos	I
Dedicatoria	II
Contenido.....	1
Índice de figuras	6
Capítulo 1. Introducción	
1.1 Introducción	9
1.2 Antecedentes	10
1.2 Definición del problema	10
1.3 Objetivo	10
1.4 Justificación	10
1.5 Hipótesis	10
1.6 Alcance del proyecto	11
1.5 Metodología de la investigación	11
Capítulo 2. Conceptos básicos	
2.1 Base de Datos	12
2.2 Modelo de datos Entidad - Relación (E - R)	12
2.2.1 Entidad	13

2.2.2 Relación(interrelación)	13
2.2.3 Cardinalidad	14
2.2.4 Atributo	14
2.2.5 Identificador	15
2.3 Sistema de Gestión de Bases de Datos (DBMS)	15
2.3.1 Redundancia e inconsistencia de datos	16
2.3.2 Dificultad para tener acceso a los datos	16
2.3.3 Aislamiento de Datos	16
2.3.4 Problemas de Seguridad	16
2.3.5 Problemas de integridad	16
2.4 Tecnologías de Acceso a Bases de Datos	17
2.4.1 Open DataBase Connectivity(ODBC)	17
2.4.2 Java DataBase Connectivity(JDBC)	17
2.4.3 ActiveX Data Objects(ADO).....	18
2.5 Cliente - Servidor	18
2.5.1 Elementos principales.....	19
2.5.2 Evolución de la arquitectura Cliente - Servidor	19
2.6 Lenguaje de Consulta Estructurado(SQL).....	19
2.6.1 Microsoft SQL Server.....	20
Capítulo 3. Conceptos básicos de bases de datos en la Web	
Introducción	21

3.1 Servidores Web.....	21
3.1.1 Principales Servidores Web	21
3.1.1.1 Apache	22
3.1.1.2 AOL Server	22
3.1.1.3 Internet Information Server (IIS)	23
3.2 Seguridad	23
3.3 Integración de Bases de Datos en el Web	24
3.4 ¿Cómo funciona la integración de bases de Datos en el Web?	24
3.5 Tecnologías para la integración de Bases de Datos en el Web.....	25
3.5.1 Common GateWay Interface(CGI)	25
3.5.2 Active Server Pages (ASP).....	26
3.5.3 Hipertext Preprocesor (PHP)	27
3.5.4 Java Server Pages (JSP)	28
3.5.5 Extensible Markup Language (XML)	29
Capítulo 4. Diseño del Sistema	
4.1 Diagramas de la Base de Datos	30
4.1.1 Diagrama Entidad - Relación	30
4.1.2 Diagrama físico	31
4.2 Diseño de las tablas de la Base de Datos	31
4.3 Diseño de Entradas.....	34
4.4 Salidas del Sistema.....	42

4.4.1 Reportes.....	42
4.4.2 Reportes en el Web	43
Capítulo 5. Implementación del Sistema	
5.1 Determinación de Requerimientos	45
5.2 Requerimientos de Hardware y Software	45
5.3 Instalación del sistema.....	46
5.3.1 Instalación del Cliente.....	46
5.3.2 Instalación del Servidor.....	49
5.3.3 Instalación del Sitio Web.....	49
5.4 Pruebas y Resultados	50
5.5 Especificaciones del modo de uso.....	51
5.5.1 Botones de acción	51
5.5.2 Teclas de función	53
5.6 Utilerías S.A.R.	53
5.6.1 Copia de seguridad	54
5.6.2 Restauración	55
5.6.3 Instalación de la Base de Datos	56

Apéndice A: Cómo crear una conexión ODBC y ADO	57
Apéndice B: Instalación de IIS en Windows 2000 y XP	66
Apéndice C: Códigos Fuente	84
Apéndice D: Introducción a la Ingeniería de Software	106
Conclusiones.....	111
Glosario.....	112
Bibliografía.....	114

ÍNDICE DE FIGURAS

Capítulo 2. Conceptos básicos

2.1 Conceptos del modelo entidad-relación	13
---	----

Capítulo 3. Conceptos básicos de bases de datos en la Web

3.1 Esquema básico de funcionamiento CGI	25
3.2 Esquema de conexión mediante software para Base de Datos	26
3.3 Esquema de funcionamiento de ASP	27
3.4 Esquema del funcionamiento de las páginas PHP.....	28

Capítulo 4. Diseño del Sistema

4.1 Diagrama Entidad-Relación S.A.R.	30
4.2 Diagrama Físico S.A.R	31
4.3 Tabla Dependencias.	31
4.4 Tabla Conectado	32
4.5 Tabla Mail	32
4.6 Tabla Servicio	32
4.7 Tabla servicios	33
4.8 Tabla tipo_user	33
4.9 Tabla Ubica	33
4.10 Tabla Tiponic	33
4.11 Tabla UserNT	33
4.12 Tabla Tips	33
4.13 Tabla Tipo_sistema	34

4.14 Alta de usuarios conectados.	34
4.15 Modulo Dependencias	35
4.16 Usuarios de NT	36
4.17 Usuarios de Correo	37
4.18 Servicios	37
4.19 Tips para la pagina Web	38
4.20 Reportes	39
4.21 Alta de conceptos.	40
4.22 Entrada al sitio WEB para consulta de datos	41
4.23 Selección de opciones para la consulta Web	41
4.24 Reporte de las Dependencias	42
4.25 Reporte de Dependencias en una hoja de Word	43
4.26 Reporte WEB de usuarios conectados	43
4.27 Reporte WEB de usuarios de correo electrónico	44
4.28 Solución a preguntas frecuentes	44

Capítulo 5. Implementación del Sistema

5.1 Inicio de la instalación	46
5.2 Pantalla de bienvenida a la instalación	47
5.3 Pantalla de elección de directorio para la instalación	47
5.4 Pantalla de opciones para la instalación	48
5.5 Proceso de instalación	48
5.6 Instalación de la Base de Datos desde el Query Analyzer	49
5.7 Pagina Web Default	50
5.8 Barra de menú	53
5.9 Pantalla de bienvenida al S.A.R.	53

5.10 Ventana principal de Utilerías	54
5.11 Pantalla para realizar copias de Seguridad	54
5.12 Pantalla para realizar la restauración de la Base de Datos	55
5.13 Cuadro de dialogo de confirmación.	56

Capítulo 1

Introducción

1.1 Introducción

A partir del surgimiento de la computadora personal, la computación como herramienta de administración y cambio ha tenido su advenimiento.

Durante décadas, las innovaciones tecnológicas en el campo del computo han tendido a mejorar su interfaz y potencia para servir mas y mejor al hombre.

Sin embargo, el humano no es un ser aislado, por ende, la conectividad e ínter conectividad son, actualmente, la coyuntura de cambio tecnológico.

Por la razón anterior las redes de computadoras han cobrado suma importancia, ya que con esta herramienta se comparten y explotan recursos tanto físicos como lógicos y permiten la interacción y las comunicaciones entre usuarios, departamentos y corporaciones sin importar la ubicación geográfica.

Es ahora, debido al momento tecnológico por el cual atraviesan las ciencias, las instituciones publicas y privadas, los sistemas computacionales y las comunicaciones debe actualizarse.

Es por ello que el Sistema Estatal de Comunicaciones Digitales(SECODI) a través de la Subdirección de Administración de Redes, departamento encargado de administrar a los usuarios de la red, decide mejorar sus procesos de consultas y reportes a cada una de las dependencias correspondientes, haciendo uso de las tecnologías Cliente – Servidor y con ello ofrecer un mejor servicio.

El Sistema de Administración de Red (S.A.R.) es un programa codificado en un lenguaje de programación visual (Delphi 6.0), ya que por su diseño y facilidad de manejo con las bases de datos facilita el diseño de la aplicación y agiliza los procesos. Para que el S.A.R. tenga acceso a la información ejecuta sentencias del Lenguaje de Consulta Estructurado (SQL) para realizar la conexión al servidor donde esta alojada la Base de datos.

Con todo ello, el S.A.R. ofrece diversas ventajas como:

- Ejecución del sistema en diversas plataformas: Windows 98, Windows Me, Windows XP, Windows 2000(profesional, Server y Advanced Server), Windows Server 2003.
- Consultas diseñadas para una salida en borrador y una salida con formato utilizando Microsoft Word.
- Consulta de datos por parte de los usuarios de una manera mas rápida y eficaz lo cual permite una mejor atención y administración de la información.
- Consulta de cada uno de los jefes de informática de cada dependencia por medio de una pagina Web.

1.2 Antecedentes

Con versiones anteriores del sistema al no tener un Manejador de Bases de Datos adecuado (DBMS) la información estaba compartida para que los usuarios accedieran directamente a ella mediante la red, aunado a esto la ejecución del Software se realizaba desde la máquina de uno de los administradores lo cual hacía que ese equipo se volviera lento.

Cuando ocurrían problemas con la red o con alguno de los equipos de los cuales dependía el software para funcionar se realizaban los reportes y las consultas necesarias se tenían que realizar manualmente lo cual era tedioso y complicado para los administradores.

1.3 Definición del problema

El software usado anteriormente para la administración requería tener ciertos privilegios en las cuentas de los usuarios que lo usaban, lo cual provocaba que la seguridad de su servidor se volviera vulnerable, debido a esto en muchas ocasiones el software no podía ser utilizado.

A pesar de ser un software diseñado para trabajar en un ambiente cliente / servidor era muy lento y se sobrecargaba debido a la utilización de varias operaciones sobre la misma tabla.

1.4 Objetivo

El S.A.R. tiene como objetivo principal la optimización y administración de los datos de los usuarios de la red corporativa del Gobierno del Estado de Hidalgo y con ello proporcionar a los usuarios finales una visión abstracta de los datos escondiendo ciertos detalles de como se almacenan y mantienen los datos para poder manipularlos y presentar información más clara y sin demoras.

Realizar la alta de equipos desde el lugar donde está situada la dependencia sin necesidad de trasladarse hasta el edificio principal para que los administradores realicen este proceso.

1.5 Justificación

Con el diseño del S.A.R. se pretende proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la Base de Datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del S.A.R., por lo que este software funciona como interfase entre los usuarios y la Base de Datos.

1.6 Hipótesis

Con estas mejoras se pretende brindar una mejor administración de los equipos conectados, usuarios de correo electrónico, y solucionar los errores que se presentaban en la versión anterior,

así mismo proporcionar a los administradores un sistema que no presente tantos inconvenientes, que su manejo sea más sencillo y sobre todo que tengan un sistema confiable.

1.7 Alcance del proyecto

La versión anterior del sistema llevaba a cabo ciertas funciones lo cual solo permitía el uso del software en el área correspondiente a los administradores, con esta nueva versión y la implementación de un sitio web pueden realizar funciones básicas de consulta y alta de usuarios que posteriormente se complementa con el uso del S.A.R. para complementar los datos correspondientes a dicho equipo.

Al sitio web en versiones posteriores se pueden anexar algunas mejor como autenticación con un certificado (SSL) emitido por alguna empresa, cifrado de contraseñas para una mayor seguridad al momento de enviar los datos, entre otras cosas.

1.8 Metodología de la Investigación

Para llevar a cabo el diseño del S.A.R. se llevaron a cabo una serie de procesos para poder obtener la información necesaria para tener una base sólida y con ello tener un mejor desempeño del software.

- Analizar una Metodología Estructurada para desarrollar el Ciclo de Vida completo de los Sistemas Informáticos (ver Apéndice D).
- Resolver algunos casos prácticos de Análisis y Diseño de Sistemas Informáticos basados en casos reales de la empresa.
- Informarse de las plataformas y arquitecturas tecnológicas disponibles y según las tendencias y estándares del mercado, con especial mención a los sistemas abiertos basados en modelos Cliente / Servidor, Sistemas de Gestión de Bases de Datos.
- Conocer el Lenguaje de Consulta Estructurado (SQL) para un mejor desempeño en almacenamiento y procesamiento de los datos.
- Realizar entrevistas con los usuarios del sistema para considerar las propuestas para mejorar ciertos puntos básicos en el S.A.R..
- Elección de entre varios lenguajes de programación el más adecuado para el manejo de Bases de Datos de manera confiable y con ello agilizar los procesos.

Capítulo 2

Conceptos Básicos

Las Bases de Datos son sistemas que sirven como archivos donde se almacena información, cuya finalidad es mantener dicha información y hacer que esté disponible cuando se solicite. Estos sistemas nos ofrecen una serie de recursos para poder realizar operaciones sobre dicha información: agregar datos, actualizarlos, eliminar, consultar, relacionarlos con otros datos entre otras cosas.

2.1 Base de Datos

Es un conjunto de datos que es utilizado por los sistemas de aplicación de alguna empresa dada. El termino de empresa es simplemente un termino genérico para la identificación de cualquier organización independiente de tipo comercial, técnico, científico u otro.

Campo, Es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto. Este puede ser de diferentes tipos de acuerdo a la información que va a contener, es decir, texto, Fecha, Entero, caracteres.

Registro, Colección de campos de iguales o de diferentes tipos.

Base de datos, Es una colección de archivos interrelacionados, son creados con un Sistema Manejador de Bases de Datos(DBMS). El contenido de una base de datos engloba a la información concerniente de una organización, de tal manera que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del manejo del sistema.

Administrador de base de datos (DBA), Es la persona o equipo de personas profesionales responsables del control y manejo del sistema de base de datos, generalmente tiene experiencia en DBMS, diseño de bases de datos, Sistemas operativos, comunicación de datos, hardware y programación.

2.2 Modelo de datos entidad-relación (E-R)

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

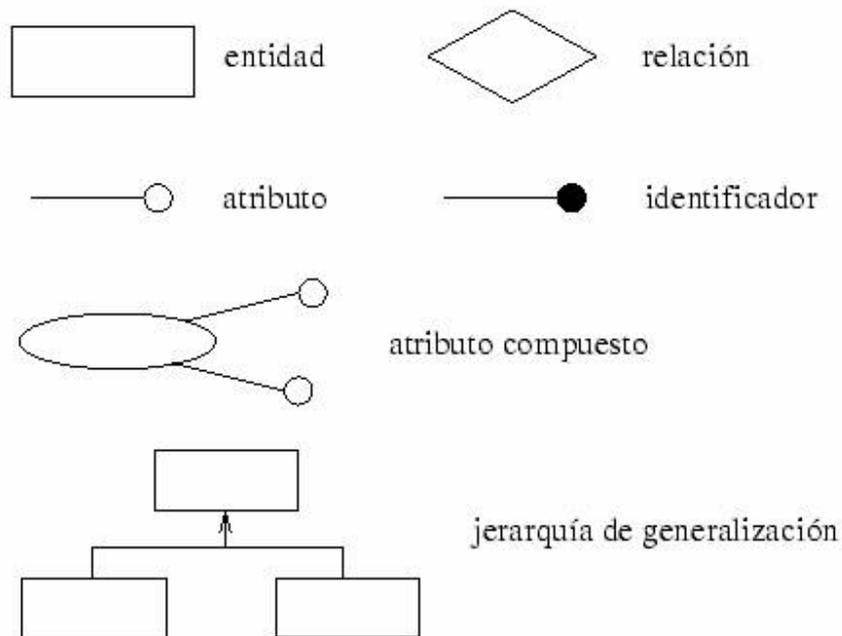


Figura 2.1: Conceptos del modelo entidad-relación

2.2.1 Entidad

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles.

- Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad.
- Una entidad fuerte es una entidad que no es débil.

2.2.2 Relación (interrelación)

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes en una relación es lo que se denomina grado de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación binaria; si son tres las entidades participantes, la relación es ternaria y así sucesivamente.

Una relación recursiva es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

2.2.3 Cardinalidad

La cardinalidad con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es obligatoria si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es opcional.

Una de las trampas que pueden encontrarse ocurre cuando el esquema representa una relación entre entidades, pero el camino entre algunas de sus ocurrencias es ambiguo. El modo de resolverla es reestructurando el esquema para representar la asociación entre las entidades correctamente.

2.2.4 Atributo

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen.

Cada atributo tiene un conjunto de valores asociados denominado dominio. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos pueden ser simples o compuestos. Un atributo simple es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un atributo compuesto es un atributo con varios componentes, cada uno con un significado por sí mismo. Un grupo de atributos se representa mediante un atributo compuesto cuando tienen afinidad en cuanto a su significado, o en cuanto a su uso.

Un atributo compuesto se representa gráficamente mediante un óvalo.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un atributo monovalente es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un atributo polivalente es aquel que tiene varios valores para cada ocurrencia de la

entidad o relación a la que pertenece. A estos atributos también se les denomina multivaluados, y pueden tener un número máximo y un número mínimo de valores. La cardinalidad de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. El valor por omisión es (1,1).

Por último, los atributos pueden ser derivados. Un atributo derivado es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

2.2.5 Identificador

Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos. Las relaciones no tienen identificadores.

2.3 Sistema de Gestión de Bases de Datos (DBMS database management system)

Un sistema de gestión de bases de datos (DBMS database management system) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos, es básicamente un sistema computarizado para llevar registros, es decir, un sistema cuya finalidad general es almacenar información y permitir a los usuarios realizar una variedad de operaciones sobre dichos registros, por ejemplo: insertar, borrar, modificar, agregar, consultar. La colección de datos se denomina base de datos (BD).

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

Los sistemas de bases de datos están diseñados para gestionar grandes bloques de información, que implica tanto la definición de estructuras para el almacenamiento como de mecanismos para la gestión de la información.

Además los DBMS deben mantener la seguridad de la información almacenada pese a la caída del sistema o accesos no autorizados.

Los objetivos principales de un sistema de base de datos es disminuir los siguientes aspectos:

2.3.1 Redundancia e inconsistencia de datos.

Puesto que los archivos que mantienen almacenada la información son creados por diferentes tipos de programas de aplicación existe la posibilidad de que si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información, es decir que la misma información sea más de una vez en un dispositivo de almacenamiento.

2.3.2 Dificultad para tener acceso a los datos.

Un sistema de base de datos debe contemplar un entorno de datos que le facilite al usuario el manejo de los mismos. El sistema debe de prever estas situaciones desde que se realiza el diseño para evitar una deficiencia.

2.3.3 Aislamiento de los datos.

Puesto que los datos están repartidos en varios archivos, y estos no pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los datos apropiados.

2.3.4 Anomalías del acceso concurrente.

Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los datos simultáneamente.

En un entorno así la interacción de actualizaciones concurrentes puede dar por resultado datos inconsistentes. Para prevenir esta posibilidad debe mantenerse alguna forma de supervisión en el sistema.

2.3.5 Problemas de seguridad.

La información de toda empresa es importante, aunque unos datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de base de datos sea confiable debe mantener un grado de seguridad que garantice la autenticación y protección de los datos.

2.3.6 Problemas de integridad.

Los valores de datos almacenados en la base de datos deben satisfacer cierto tipo de restricciones de consistencia. Estas restricciones se hacen cumplir en el sistema añadiendo códigos apropiados en los diversos programas de aplicación.

2.4 Tecnologías de acceso a Base de Datos

2.4.1 Open DataBase Connectivity (ODBC)

Open DataBase Connectivity (ODBC) es un intermediario entre bases de datos y aplicaciones, cuya tarea es sostener una conversación de preguntas y respuestas entre dos "sujetos" que no hablan el mismo idioma y que gestionan sus recursos de forma diferente.

En el ambiente Windows, Microsoft creó la tecnología ODBC pensando en este problema. ODBC es un armatoste que alberga controladores. El armatoste sirve para gestionar los controladores, y los controladores son los que saben "hablar" con las bases de datos.

La mayoría de software para Windows diseñado para el uso de bases de datos, utilizan para comunicarse el ODBC; y los fabricantes de bases de datos han diseñado los controladores de sus sistemas para ponerlos el ODBC, para que de esta manera otras aplicaciones puedan acceder a la información contenida en los programas manejadores de Bases de Datos (DBMS).

En ODBC solo realiza una simple tarea, crear un origen de datos. Un origen de datos consiste en el nombre, el controlador y la base de datos. (Ver apéndice A).

Las aplicaciones creadas específicamente para Windows 9X, 2000 y NT usan el ODBC de 32 bits; pero algunos sistemas conservan un ODBC de 16 bits para las aplicaciones de con versiones anteriores.

2.4.2 Java DataBase Connectivity (JDBC)

Java DataBase Connectivity (JDBC), le da una habilidad a la programación en JAVA para integrarse con DBMS comerciales y con ello es posible crear un ambiente ideal tipo Cliente / servidor.

Hoy en día, la mayoría de los gestores de bases de datos relacionales tienen soporte para la utilización de interfaces Web, con ello favorecen a la interfaz, pero en aplicaciones más complejas se presentan ciertas limitaciones que no permiten generar un buen trabajo. En estos casos, es conveniente la utilización de lenguajes de programación como Delphi, que permitan elaborar aplicaciones para generar el interfaz con la base de datos.

Ahora definiremos lo que es JDBC. Es una Aplicación de JAVA para permitir ejecutar instrucciones SQL (Lenguaje estructurado de consulta), que es un lenguaje de alto nivel para crear, manipular, examinar y gestionar bases de datos relacionales.

Podemos decir en tres frases lo que hace JDBC:

- Establece una conexión con una BD, que puede ser remota o no.
- Envía sentencias SQL a la BD.

- Procesa los resultados obtenidos de la BD.

JDBC esta incluido dentro del lenguaje Java para el acceso a bases de datos. Consiste en un conjunto de clases e interfaces escritos en Java que ofrecen un completo conjunto de herramientas para la programación de bases de datos, por lo tanto es la una solución que permite el acceso a bases de datos.

2.4.3 ActiveX Data Objects (ADO)

ActiveX Data Objects (ADO) es una tecnología ampliable y de fácil uso para agregar a sus páginas Web acceso a bases de datos. Puede utilizar ADO para escribir secuencias de comandos compactas y escalables que le permitan conectarse a orígenes de datos, como bases de datos, hojas de cálculo, archivos de datos secuenciales o directorios de correo electrónico.

OLE DB es una interfaz de programación de nivel de sistema que proporciona un conjunto estándar de interfaces para que permitan exponer las funciones del sistema de administración de bases de datos. Con el modelo de objetos ADO es fácil tener acceso a estas interfaces para agregar funciones de bases de datos a las aplicaciones Web. Además, también puede utilizar ADO para tener acceso a bases de datos compatibles con la Conectividad abierta de bases de datos (ODBC, Open Database Connectivity).

La conectividad con bases de datos, requiere de una sintaxis muy es sencilla y fácil de utilizar. (Ver apéndice A sección 2)

2.5 Cliente – Servidor

La programación cliente-servidor se utiliza cuando queremos realizar aplicaciones que utilicen redes y que comuniquen entre sí a varios equipos.

Con respecto a la definición de arquitectura cliente / servidor se encuentran las siguientes definiciones:

- Cualquier combinación de sistemas que pueden colaborar entre si para dar a los usuarios toda la
- información que ellos necesiten sin que tengan que saber donde esta ubicada.
- Es una arquitectura de procesamientos cooperativo donde uno de los componentes pide servicios a otro.
- Es un procesamiento un procesamiento de datos de índole colaborativo entre dos o más computadoras conectadas a una red.

El termino cliente / servidor es originalmente aplicado a la arquitectura de software que describe el procesamiento entre dos o mas programas: una aplicación y un servicio soportante.

2.5.1 Elementos principales

Los elementos principales de la arquitectura cliente / servidor son justamente el elemento llamado cliente y el otro elemento llamado servidor.

Básicamente consiste en que el programa se divide en dos partes:

- La parte Cliente, que reside en el equipo donde está el usuario y se encarga de la interacción con éste.
- La parte Servidor, que reside en un ordenador conectado a la red permanentemente y se encarga de manipular los datos.

Ambas partes de la aplicación se comunican entre sí utilizando algún protocolo de red TCP/IP.

2.5.2 Evolución de la arquitectura cliente / servidor

La era de la computadora central desde sus inicios el modelo de administración de datos a través de computadoras se basamento en el uso de terminales remotas, que se conectaban de manera directa a una computadora central. Dicha computadora central se encargaba de prestar servicios caracterizados por que cada servicio se prestaba solo a un grupo exclusivo de usuarios. El personal de la llamada área de sistemas se encargaba de consolidar o integrar la información cuando las necesidades de los usuarios lo exigían.

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores estos últimos responden a la demanda del cliente que la produjo. Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet.

Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura.

2.6 Lenguaje de Consulta Estructurado (SQL)

Desarrollado por los laboratorios de IBM durante 1974-1978, es un lenguaje autosuficiente y siempre está asociado al Gestor de la Base de Datos (DBMS). Dependiendo del fabricante podremos encontrar el SQL como manipulador de las Bases de Datos en INFORMIX, ORACLE, ACCESS, entre otras. El SQL permite reducir el tiempo requerido para:

- Organizar
- Almacenar

- Recuperar
- Sumarizar
- Agrupar
- Formatear

El SQL está dividido en tres componentes principales:

- **DML - Data Manipulation Language** (Lenguaje de Manipulación de Datos): Contiene los componentes del lenguaje para la manipulación de la información almacenada en las tablas.
- **DCL - Data Control Language** (Lenguaje de Control de Datos): Incluye la sintaxis comúnmente utilizada para definir, modificar o borrar las tablas en las que se almacenan los datos.
- **DDL - Data Definition Language** (Lenguaje de Definición de Datos): Contiene elementos útiles para trabajar en un entorno multiusuario en el que es importante la protección de la información, la seguridad de las tablas y el establecimiento de restricciones de acceso.

2.6.1 Microsoft SQL Server

SQL Server es el servidor de bases de datos de Microsoft, seguro, robusto y con las más avanzadas prestaciones: transacciones, procedimientos almacenados, etc. Además podrá gestionar sus bases de datos en nuestros servidores utilizando la consola de SQL Server que tenga instalada en el equipo de trabajo (servidor con plataforma Windows 2000 o Windows XP profesional).

SQL Server realiza tareas de una manera más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa que tiene un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.

Capítulo 3

Conceptos básicos de Bases de Datos en la Web

Introducción

El Web es un medio para localizar, enviar, recibir información de diversos tipos, aun con las bases de datos. En el ámbito competitivo, es esencial ver las ventajas que esta vía electrónica proporciona para presentar la información, reduciendo costos y el almacenamiento de la información, y aumentando la rapidez de difusión de la misma.

En el pasado, las bases de datos sólo podían utilizarse al interior de las instituciones o en redes locales, pero actualmente el Web permite acceder a bases de datos desde cualquier parte del mundo. Estas ofrecen, a través de la red, un manejo dinámico y una gran flexibilidad de los datos, como ventajas que no podrían obtenerse a través de otro medio informativo.

Con estos propósitos, los usuarios de Internet o Intranet pueden obtener un medio que puede adecuarse a sus necesidades de información, con un costo, inversión de tiempo, y recursos mínimos. Asimismo, las bases de datos serán usadas para permitir el acceso y manejo de la variada información que se encuentra a lo largo de la red.

3.1 Servidores Web

El servidor de Páginas es la parte primordial de cualquier sitio de Internet, ya que es el encargado de generar y enviar la información a los usuarios finales.

Sin embargo conforme las demandas de los Servidores de paginas Web incrementaron fue necesario agilizar este proceso, ya que el llamar un interpretador para que ejecutara otro programa ponía una demanda muy fuerte sobre el equipo que mantenía el Servidor de Paginas Web.

Hoy en día, se pudiera decir que todos los Servidores de Paginas Web son Servidores de Aplicaciones ya que se les ha desarrollado alguna funcionalidad especial que les permite realizar, valga la redundancia, aplicaciones de Servidor .

Dependiendo de la funcionalidad se trae consigo complejidad al sistema, ya sea en la forma de requerimientos del sistema (memoria , procesadores), carga administrativa (configuración, tiempo de desarrollo) o alguna otra.

3.1.1 Principales Servidores Web

Entre los principales Servidores Web podemos encontrar los siguientes:

- Apache
- Nestcape Enterprise 3.0
- Microsoft IIS
- WebLogic Tengah
- Lotus Domino Go Web Server
- IBM Internet Conecction Server
- Java Web Server

A continuación se dará una explicación de algunos de los Servidores Web antes mencionados.

3.1.1.1 Apache

Apache es uno de los Servidores de paginas más utilizados, posiblemente porque ofrece instalaciones sencillas para sitios pequeños y si se requiere es posible expandirlo hasta el nivel de los mejores productos comerciales. Si se utiliza para un sitio pequeño que solo contenga archivos en HTML, esto es, no requiera de aplicaciones de servidor su funcionalidad es excelente.

Cuando el Servidor de Paginas Web recibe la requisición de un cliente, éste reconoce cuando debe enviar un documento estático o ejecutar algún tipo de aplicación, si se invoca un programa en Perl y este a su vez solicita información a una base de datos, por lo tanto para llevar acabo esta operación se inician dos procesos nuevos, y si no se tienen los suficientes recursos en cuanto a memoria y procesadores se refiere, seguramente el servidor será insuficiente.

Apache tiene tanto tiempo de desarrollo que han sido desarrolladas diferentes soluciones para evitar ineficiencias como la antes mencionada, algunas:

- Es capaz de utilizar otros interpretadores y lenguajes como PhP, Python.
- Puede conectarse directamente a una Base de datos .

3.1.1.2 AOLServer

AOLServer fue diseñado conociendo varias deficiencias que existían en el modelo inicial utilizado por Apache, desde sus versiones iniciales fue desarrollado con Threading en mente, esto es, compartir la memoria del Proceso general en varios sub-procesos, esto no solo vuelve eficiente las conexiones al servidor de paginas sino también reduce la carga sobre el mismo.

Otra ventaja de AOLServer es el ofrecimiento de Aol Dynamic Pages (ADP) que son muy similares a las Active Server Pages(ASP) de Microsoft o Java Server Pages(JSP) por Java.

3.1.1.3 Internet Information Server(IIS)

Internet Information Server es el servidor de páginas desarrollado por Microsoft para Windows, a diferencia de los dos servidores de páginas mencionados anteriormente, IIS solo puede operar en plataformas Windows.

IIS engloba un conjunto de herramientas destinadas al control de servicios de Internet como el Web, FTP, correo y servidores de noticias. Además incluye el soporte necesario para la creación de páginas dinámicas en el servidor.

IIS otorga muchas posibilidades en cuanto a los servicios de Internet crear y administrar sitios web rápidos y seguros los cuales pueden ofrecer a las empresas el mejor servicio de información con las herramientas que proporciona el IIS (Microsoft Internet Information Server).

IIS es una de herramienta fácil de utilizar la cual permite realizar de manera sencilla y practica algunas cosas como son:

- Montar un servidor web de los más utilizados: instalar Internet Information Server; o bien instalar Windows 2000 Server.
- Configurar el servidor ftp. Manejar los registros generados por los servidores.
- Qué es un servidor SMTP. Configuración del servidor de correo. Qué es un servidor NNTP y su configuración.
- Configuración del servidor de DNS: zonas, registros de recursos y delegación de zonas.
- Usar aplicaciones de contenido dinámico. Acceso a bases de datos con Windows 2000 Server e IIS. Aplicaciones CGI.
- Notas generales de seguridad. Sistemas de autenticación en Internet Information Server y control de acceso. Certificados de seguridad.

3.2 Seguridad

La evaluación de este punto es uno de los más importantes en la interconexión del Web con bases de datos. A nivel de una red local, se puede permitir o impedir, a diferentes usuarios el acceso a cierta información, pero en la red mundial de Internet se necesita de controles más efectivos en este sentido, ante posible espionaje, copia de datos, manipulación de éstos, entre otras cosas.

Para la integración de bases de datos con el Web es necesario contar con una interfaz que realice las conexiones, extraiga la información de la base de datos, le dé un formato adecuado de tal manera que puede ser visualizada desde un navegador de Web, y permita lograr sesiones interactivas entre ambos, dejando que el usuario haga elecciones de la información que requiere

3.3 Integración de Bases de Datos en el Web

Gracias a la conexión con la red mundial Internet, poco a poco, cada individuo o institución va teniendo acceso a mayor cantidad de información de las diversas ramas de la ciencia con distintos formatos de almacenamiento.

La mayor parte de información es presentada de forma estática a través de documentos HTML, lo cual limita el acceso a los distintos tipos de almacenamiento en que ésta pueda encontrarse.

Pero, en la actualidad surge la posibilidad de utilizar aplicaciones que permitan acceder a información de forma dinámica, tal como a bases de datos, con contenidos y formatos muy diversos.

Una de las ventajas de utilizar el Web para este fin, es que no hay restricciones en el sistema operativo que se debe usar, permitiendo la conexión entre sí, de las páginas Web desplegadas en un Navegador Web que funciona en una plataforma, con servidores de bases de datos alojados en otra plataforma. Además, no hay necesidad de cambiar el formato o estructura de la información dentro de las bases de datos.

3.4 ¿Cómo Funciona la Integración de Bases de Datos en el Web?

Para realizar una requisición de acceso desde el Web hasta una base de datos no sólo se necesita de un navegador Web y de un Servidor Web, sino también de un software de procesamiento (aplicación CGI), el cual es el programa que es llamado directamente desde un documento HTML en el cliente. Dicho programa lee la entrada de datos desde que provienen del cliente y toma cierta información de variables de ambiente. El método usado para el paso de datos está determinado por la llamada CGI.

Una vez se reciben los datos de entrada, el software de procesamiento los prepara para enviarlos a la interfaz en forma de SQL (delphi), y luego ésta procesa los resultados que se extraen de la base de datos.

La interfaz contiene las especificaciones de la base de datos necesarias para traducir las solicitudes enviadas desde el cliente, a un formato que sea reconocido por dicha base. Además, contiene toda la información, estructuras, variables y llamadas a funciones necesarias.

El software de acceso permite el acceso a través de solicitudes con formato. Luego, el software de acceso recibe los resultados de la base de datos, aún los mensajes de error, y los pasa hacia la interfaz, y ésta a su vez, los pasa hasta el software de procesamiento.

Una aplicación que posibilita interconectar al Web con una base de datos tiene muchas ventajas, además de que las funciones que cumplen actualmente los Servidores Web y las herramientas de desarrollo de aplicaciones Web, hacen más fácil que nunca la construcción de aplicaciones más robustas. Tal vez el mayor beneficio del desarrollo de estas aplicaciones en el Web sea la habilidad de que sean para múltiples plataformas, sin el costo de distribuir múltiples versiones del software.

3.5 Tecnologías para la Integración de Bases de Datos en el Web

Cuando se utiliza una interfaz para lograr la integración del Web con cierta base de datos, se puede verificar que los procesos seguidos varían, dependiendo de la tecnología que se esté utilizando.

3.5.1 Common Gateway Interface (CGI)

Actualmente, ésta es la solución que más se está utilizando para la creación de interfaces Web/DBMS. Fue probada por primera vez en el servidor NCSA.

Se ha comprobado que si el Servidor Web recibe un URL con una llave, para devolver un documento HTML como respuesta, tendrá que cargar el servicio (programa) que le indique las variables de ambiente y de la forma HTML. La mayoría de las veces dicha llave es el "cgi-bin".

Entre las ventajas de la programación CGI, se tiene su sencillez, ya que es muy fácil de entender, además de ser un lenguaje de programación independiente, ya que los escritos CGI pueden elaborarse en varios lenguajes.

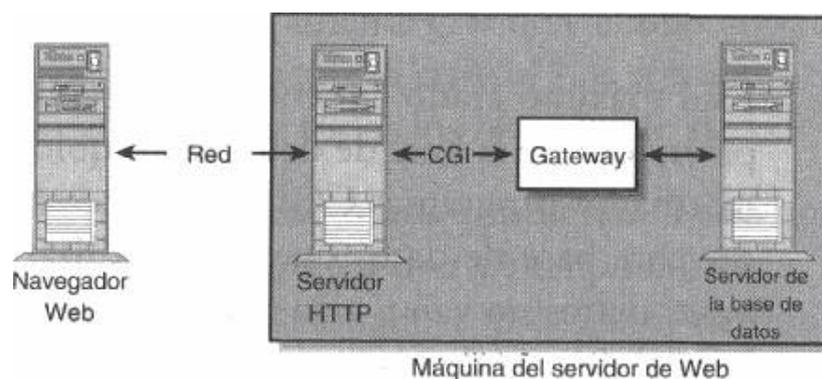


Figura 3.1 Esquema básico de funcionamiento CGI

También es un estándar para usarse en todos los servidores Web, y funcionar bajo una arquitectura independiente, ya que ha sido creado para trabajar con cualquier arquitectura de servidor Web.

Como la aplicación CGI se encuentra funcionando de forma independiente, no pone en peligro al servidor, en cuanto al cumplimiento de todas las tareas que éste se encuentre realizando, o al acceso del estado interno del mismo.

Pero el CGI presenta cierta desventaja en su eficiencia, debido a que el Servidor Web tiene que cargar el programa CGI y conectar y desconectar con la base de datos cada vez que se recibe una requisición. Además, no existe un registro del estado del servidor, sino que todo hay que hacerlo manualmente.

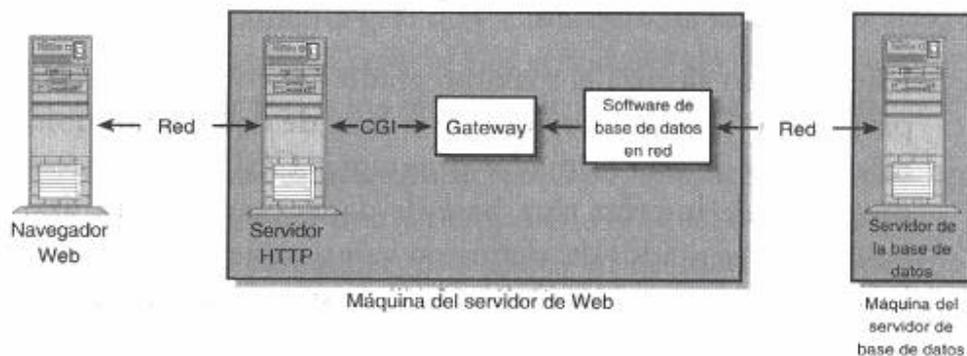


Figura 3.2 Esquema de conexión mediante software para Base de Datos

CGI es una norma para establecer comunicación entre un servidor web y un programa, de tal modo que este último pueda interactuar con navegador Web.

CGI es un programa que se ejecuta en tiempo real en un Servidor Web en respuesta a una solicitud de un Navegador Web. Cuando esto sucede el Web Server ejecuta un proceso hijo que lo recibirá los datos que envía el usuario, pone a disposición del mismo algunos datos en forma de variables de ambiente y captura la salida del programa para enviarlo como respuesta al Navegador Web.

El propósito de los CGI's es proveer inteligencia e interactividad a un sitio web, ya que nos provee de una respuesta hecha a la medida de nuestra consulta.

Un CGI se puede programar en cualquier lenguaje soportado por el sistema operativo del servidor, se debe asegurar de compilarse en modo 32 bits, de otro modo no funcionarán.

3.5.2 Active Server Pages(ASP)

Active Server Pages (ASP) es una tecnología desarrollada por Microsoft para crear páginas web de contenido dinámico apoyándose en scripts ejecutados en el servidor Web. Básicamente una página ASP es una mezcla entre una página HTML y un programa que da como resultado una página HTML que es enviada al navegador Web.



Figura 3.3 Esquema de funcionamiento de ASP

ASP es una tecnología que pertenece a la parte servidor, por esto no es necesario que el cliente o navegador la soporte ya que se ejecuta en el servidor, sí que deberemos buscar un servidor que nos soporte este tipo de tecnología para que nuestras páginas corran correctamente.

Hay que destacar que ASP es una tecnología propietaria de Microsoft, y que el uso de esta tecnología implica el uso de los productos de Microsoft (Internet Information Server y Windows) en el servidor.

Debemos considerar que por lo tanto que a la hora de programar en ASP todas las operaciones que se van a realizar, se van a efectuar siempre en el servidor y nunca en la máquina cliente.

Una página ASP es un archivo dentro de nuestro servidor de Web, compuesto por una mezcla de HTML y códigos que se ejecutarán en el servidor. Estos códigos procesarán los datos provenientes del navegador Web y generará una página Web como resultado que posteriormente se enviará a dicho cliente.

3.5.4 Hipertext Preprocesor (PHP)

Hipertext Preprocesor (PHP) es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al navegador Web. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

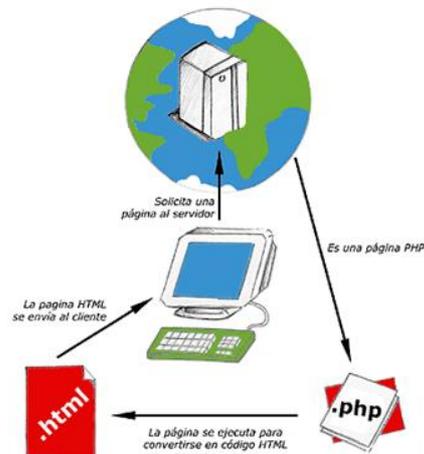


Fig. 3.4 Esquema del funcionamiento de las páginas PHP.

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad.

Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red.

3.5.4 Java Server Pages(JSP)

Java Server Pages (JSP) es una tecnología orientada a crear páginas web con programación en Java. La tecnología Java para la creación de páginas web con programación en el servidor.

Con JSP se pueden crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java.

El motor de las páginas JSP está basado en los Servlets de Java que no es otra cosa que programas en Java destinados a ejecutarse en el servidor.

En JSP se crean páginas de manera parecida a como se crean en ASP o PHP.

3.5.5 Extensible Markup Language (lenguaje de marcado extensible) XML

XML es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores.

XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Así pues, el XML juega un papel importantísimo en este mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología que permitirá compartir la información de una manera segura, fiable, fácil. Además, XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

XML no está sólo, sino que hay un mundo de tecnologías alrededor de él, de posibilidades, maneras más fáciles e interesantes de trabajar con los datos y, en definitiva, un avance a la hora de tratar la información, que es en realidad el objetivo de la informática en general. XML no es un lenguaje, sino varios lenguajes, no es una sintaxis, sino varias y no es una manera totalmente nueva de trabajar, sino una manera más refinada que permitirá que todas las anteriores se puedan comunicar entre sí sin problemas.

El XML se puede usar para infinidad de trabajos y aporta muchas ventajas en amplios escenarios. Veamos algunas ventajas del XML en algunos campos prácticos.

- Comunicación de datos. Si la información se transfiere en XML, cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
- Migración de datos. Si tenemos que mover los datos de una base de datos a otra sería muy sencillo si las dos trabajasen en formato XML.
- Aplicaciones web. Hasta ahora cada navegador interpreta la información a su manera y los programadores del web tenemos que hacer unas cosas u otras en función del navegador del usuario. Con XML tenemos una sola aplicación que maneja los datos y para cada navegador o soporte podremos tener una hoja de estilo o similar para aplicarle el estilo adecuado.

Capítulo 4

Diseño del Sistema

4.1 Diagramas de la Base de Datos

Después de realizar un análisis de toda la información obtenida con los Administradores y probables usuarios del sistema, se diseñó la Base de Datos cubriendo con esto todas las expectativas necesarias para un óptimo desempeño del S.A.R. y con ello una mejor función para los usuarios.

4.1.1 Diagrama Entidad - Relación

La figura 4.1 se ilustra el diagrama de las relaciones entre las tablas que intervienen en el S.A.R..

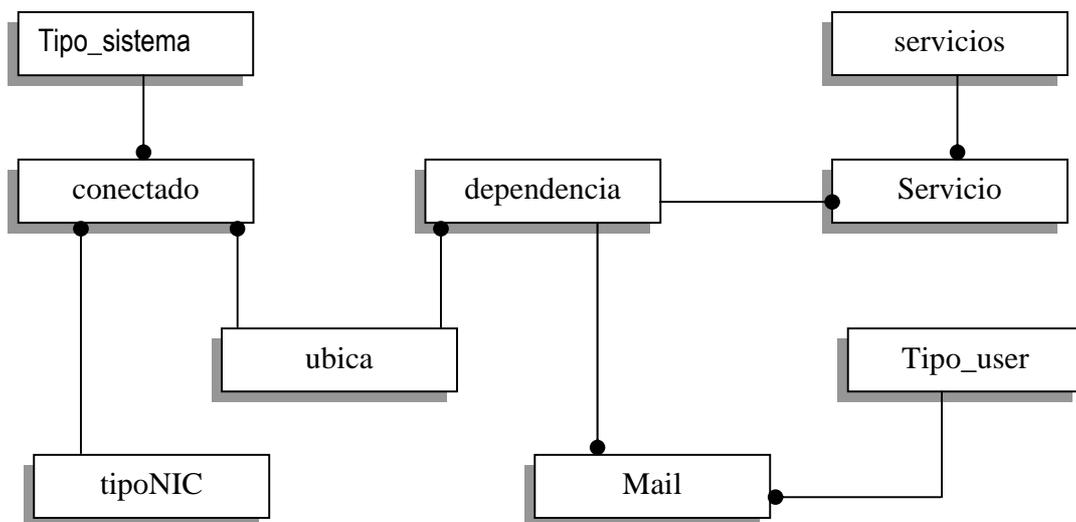


Figura 4.1 Diagrama Entidad-Relación S.A.R..

4.1.2 Diagrama físico

La figura 4.2 se ilustra el diagrama físico que se encuentra el servidor de Bases de Datos para la ejecución del S.A.R..

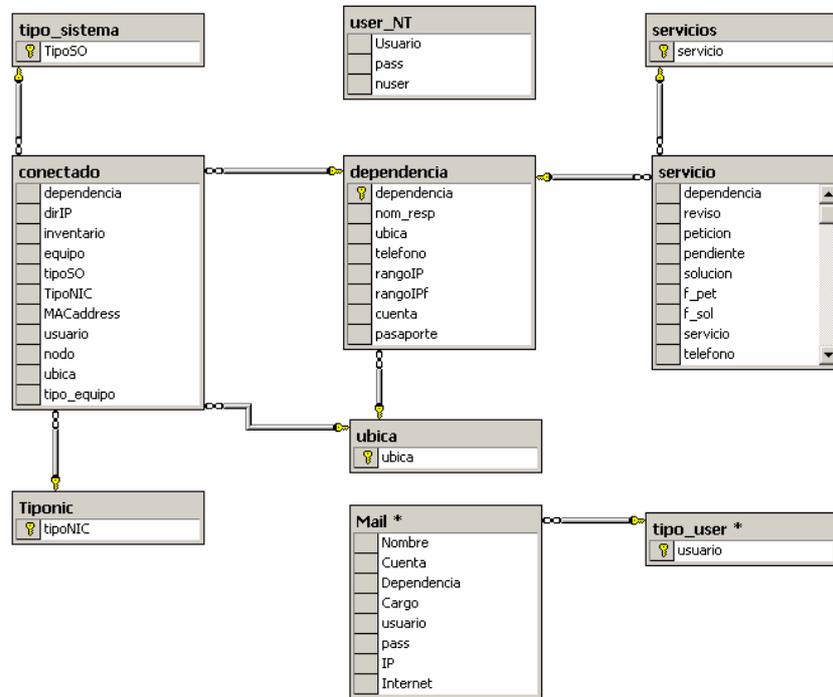


Figura 4.2 Diagrama Físico S.A.R..

4.2 Diseño de las Tablas de la Base de Datos

El S.A.R. utiliza las siguientes tablas para la administración de la información.

La tabla 4.3 se utiliza para el registro de los datos que identifican a las dependencias: Dependencia, Nom_Resp, RangolP, etc.

	Column Name	Data Type	Length	Allow Nulls
	dependencia	char	50	
	nom_resp	char	45	✓
	ubica	char	35	
	telefono	char	15	✓
	rangoIP	char	15	✓
	rangoIPf	char	15	✓
	cuenta	char	10	✓
	pasaporte	char	8	✓

Tabla 4.3 Dependencias.

En la tabla 4.4 contiene información que identifica a todos los usuarios que están conectados a la red.

Column Name	Data Type	Length	Allow Nulls
dependencia	char	50	
dirIP	char	15	✓
inventario	char	10	✓
equipo	char	100	✓
tipoSO	char	35	
TipoNIC	char	35	
MACAddress	char	20	✓
usuario	char	50	
nodo	char	3	✓
ubica	char	35	
tipo_equipo	char	30	

Tabla 4.4 Conectado.

En la tabla 4.2.3 Se tienen las columnas necesarias para identificar a los usuarios que cuentan con una cuenta de correo electrónico.

Column Name	Data Type	Length	Allow Nulls
Nombre	char	50	✓
Cuenta	char	20	✓
Dependencia	char	50	
Cargo	char	50	✓
usuario	char	50	
pass	char	15	✓
IP	char	15	✓
Internet	char	2	✓

Tabla 4.5 Mail.

La tabla 4.6 maneja la información sobre los servicios que son atendidos por el personal a cargo, los cuales pueden ser desde asesoría telefónica hasta el traslado al lugar de donde esta situada la persona para la solución.

Column Name	Data Type	Length	Allow Nulls
dependencia	char	50	
reviso	char	25	✓
peticion	text	16	✓
pendiente	text	16	✓
solucion	text	16	✓
f_pet	datetime	8	
f_sol	datetime	8	
servicio	char	20	
telefono	char	2	
No_serv	int	4	✓

Tabla 4.6 Servicio.

En la tabla 4.7 se encuentra el registro de los tipos de servicios que se ofrecen para la atención a usuarios.

	Column Name	Data Type	Length	Allow Nulls
?	servicio	char	20	

Tabla 4.7 servicios.

En la tabla 4.8 se registran los nombres de cada uno de los tipos de puestos que existen.

	Column Name	Data Type	Length	Allow Nulls
?	usuario	char	50	

Tabla 4.8 Tipo_user.

La tabla 4.9 se utiliza para obtener los datos de las ubicaciones de las dependencias.

	Column Name	Data Type	Length	Allow Nulls
?	ubica	char	35	

Tabla 4.9 Ubica.

En la tabla 4.10 se almacenan los diferentes tipos de tarjetas de red que pueden tener instalados los equipos.

	Column Name	Data Type	Length	Allow Nulls
?	tipoNIC	char	35	

Tabla 4.10 Tiponic.

La tabla 4.11 se utiliza para el registro de los usuarios que utilizan sistema operativo NT.

	Column Name	Data Type	Length	Allow Nulls
	Usuario	char	45	✓
	pass	char	15	✓
	nuser	char	15	✓

Tabla 4.11 User_NT.

En la tabla 4.12 es usada para el almacenamiento de la solución a las preguntas frecuentes que se consultan desde la página Web.

	Column Name	Data Type	Length	Allow Nulls
	Tip	text	16	
	Titulo	char	50	

Tabla 4.12 Tips.

La tabla 4.13 guarda os diferentes tipos de sistemas operativos que pueden tener instalados los equipos.

	Column Name	Data Type	Length	Allow Nulls
?	TipoSO	char	35	

Tabla 4.13 Tipo_sistema.

4.3 Diseño de Entradas

El sistema requiere de muchas entradas, en esta sección trataremos algunas de las entradas que son requeridas para el sistema.

Para el registro de usuarios o conceptos estos son dados de alta desde el sistema. Para realizar la alta de usuarios conectados, el sistema presenta una pantalla como la que se muestra la figura 4.14.

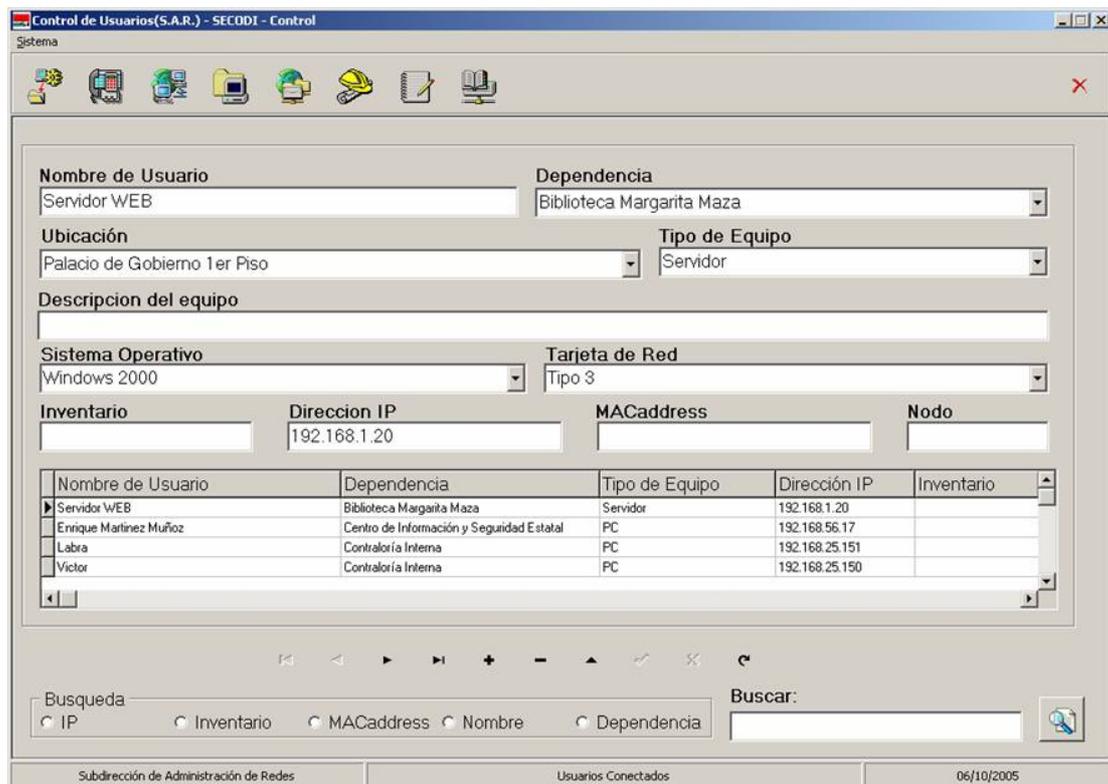


Figura 4.14 Alta de usuarios conectados.

En donde se le asigna la dependencia a la que pertenece, así mismo permite la captura del nombre completo del usuario, Ubicación donde se encuentra, ya que en algunas ocasiones aunque pertenezcan a la misma dependencia la ubicación puede variar, Tipo de Equipo, Descripción del equipo, Sistema operativo, Tarjeta de red que este usando, Inventario del equipo, Dirección IP asignada, MAC Address, Nodo al que pertenece.

Para mostrar este modulo el código se detalla en la sección 2 del Apéndice C.

La sección de búsquedas se realiza mediante la aplicación directa de filtros a las tablas, el código lo podemos ver en la sección 3 del Apéndice C.

El modulo denominado *Dependencias* nos permite dar altas de todas las dependencias que forman parte del Gobierno del Estado. Aquí se asignan las cuentas de usuario y pasaporte que serán usadas en la página Web para consultas de los equipos conectados de cada Dependencia. La figura 4.15 nos muestra la pantalla del modulo y la sección 4 del apéndice C nos muestra el código necesario para realizar las búsquedas. La sección 5 muestra el código para mostrar este modulo.

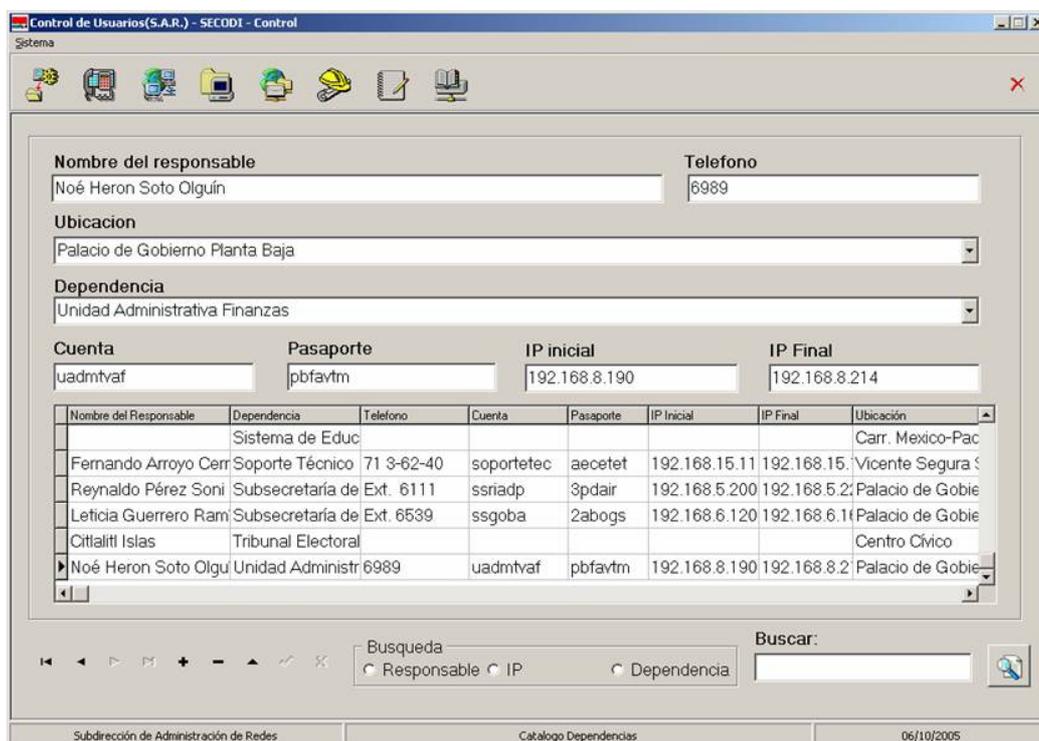


Figura 4.15 Modulo Dependencias

La figura 4.16 nos muestra la ventana correspondiente al modulo de Usuarios de sistema Operativo NT. Donde se anota el nombre del usuario y pasaporte correspondiente a cada equipo.

En el apéndice C sección 6 observamos el código correspondiente a esta búsqueda y la sección 7 la forma en que se va a mostrar este modulo.

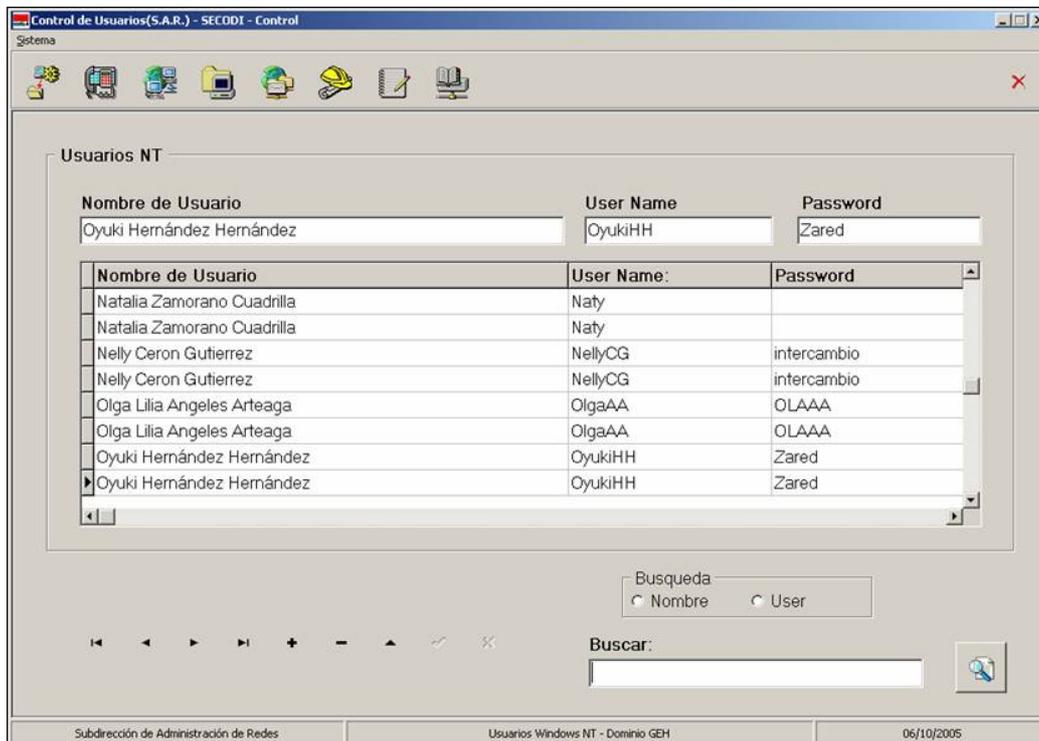


Figura 4.16 Usuarios de NT

El modulo Correo es uno de los mas importantes, ya que aquí se encuentran todos los usuarios que tienen asignada una cuenta de correo electrónico, además de su dirección Ip, el tipo de usuario(funcionario, personal de confianza, etc.), si cuentan con Internet en su equipo asignado y el nombre de usuario y pasaporte de cada cuenta.

La figura 4.17 nos muestra la ventana del modulo de Usuarios de Correo.

Para mostrar este modulo el código se detalla en la sección 9 del Apéndice C.

La sección de búsquedas se realiza mediante la aplicación directa de filtros a las tablas, el código lo podemos ver en la sección 8 del Apéndice C.

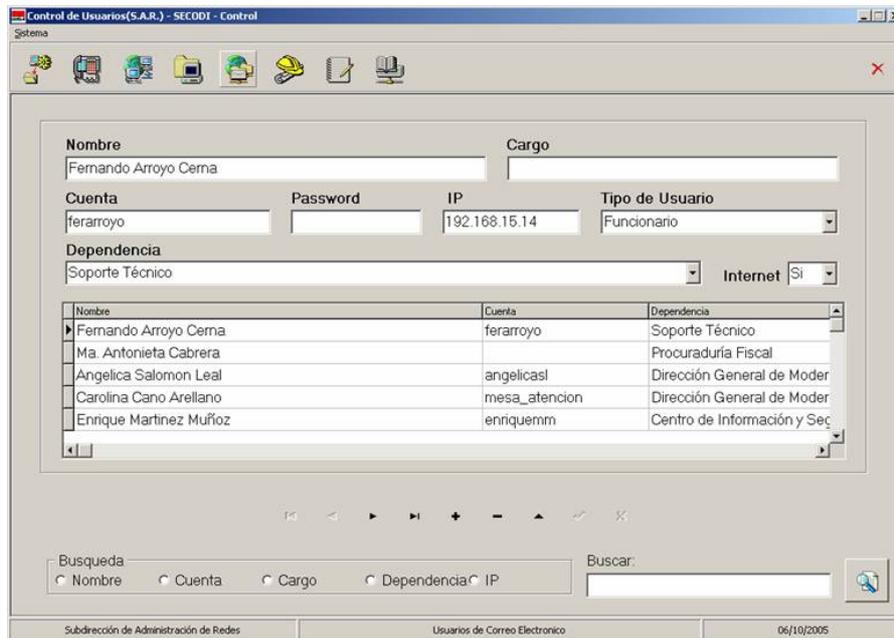


Figura 4.17 Usuarios de Correo

La figura 4.18 nos muestra la pantalla del modulo de Servicios, es usado como una bitácora donde se van anotando los servicios que se van prestando a cada una de las dependencias. Para mostrar este modulo el código se detalla en la sección 10 del Apéndice C. La sección de búsquedas se realiza mediante la aplicación directa de filtros a las tablas, el código lo podemos ver en la sección 11 del Apéndice C.

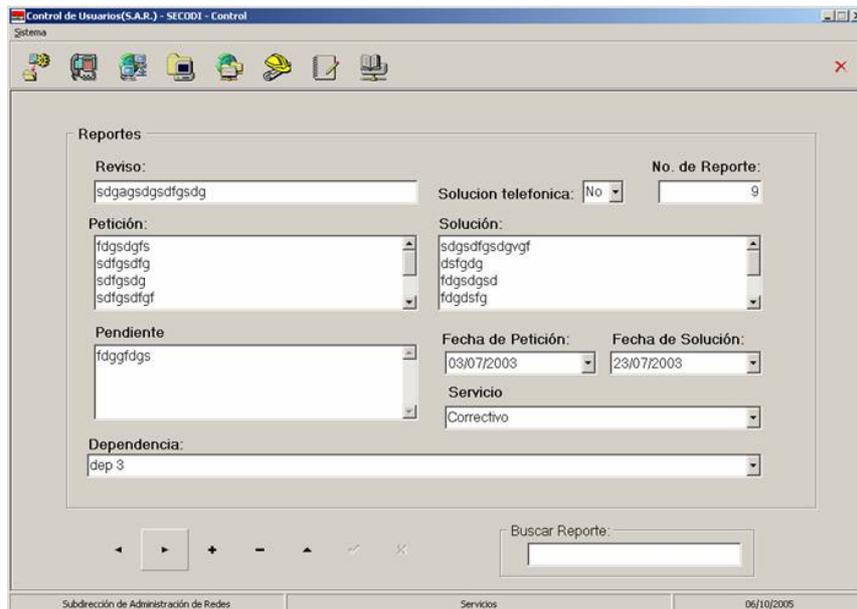


Figura 4.18 Servicios

El modulo Preguntas frecuentes (figura 4.19) es usado directamente con la pagina Web ya que como ya es común en las paginas Web existe una sección de preguntas frecuentes y mediante este botón se actualiza directamente esa sección, no es necesario modificar directamente el sitio, es decir, la pagina Web donde el usuario observa esta parte, esto hace mas fácil de mantener al día este tipo de tips o dudas a los usuarios.

En el apéndice C sección 12 observamos el código correspondiente para leer una archivo de texto para que este sea cargado en la parte de Texto y la sección 13 la forma en que se va a mostrar este modulo.

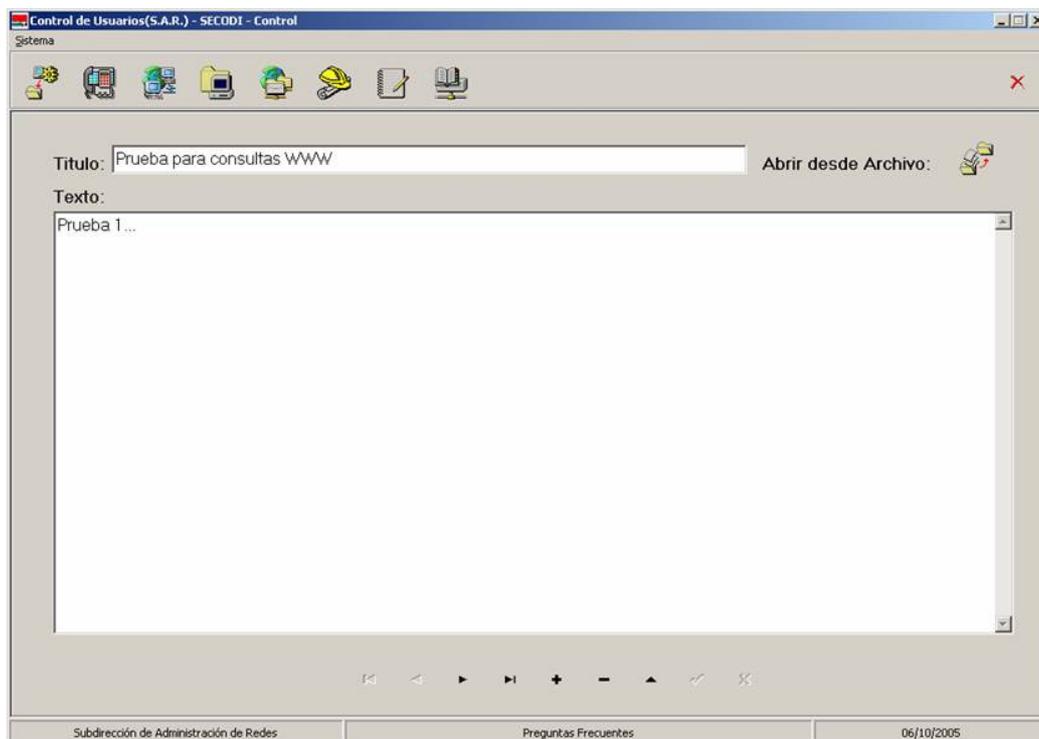


Figura 4.19 Tips para la pagina Web.

En modulo mas importante dentro del S.A.R. es el que vamos a describir a continuación y que observamos la pantalla en la figura 4.20 ya que mediante este se realizan las consultas necesarias para obtener los reportes de acuerdo a las necesidades del usuario.

Los reportes pueden ser generados en pantalla como una simple consulta además de que se puede imprimir o enviarlos directamente a un archivo de Microsoft Word con un formato previamente diseñado para ser entregados a las dependencias o a la persona que necesite esta información.

Para mostrar este modulo el código se detalla en la sección 14 del Apéndice C.

La sección de Opciones nos modifica el titulo, subtítulo y dominio de correo, el código lo podemos ver en la sección 15 para el botón Modificar y 16 para el botón Aceptar del Apéndice C.

Para la elección de opciones el código que indica un cambio en la parte derecha se la pantalla se muestra en el apéndice C sección 17.

Los dos botones que están en el centro de la ventana sirven para indicar el tipo de reporte que deseamos si lo deseamos en borrador o en la hoja de Word. El código lo observamos en la sección 17 y 18 del apéndice C.

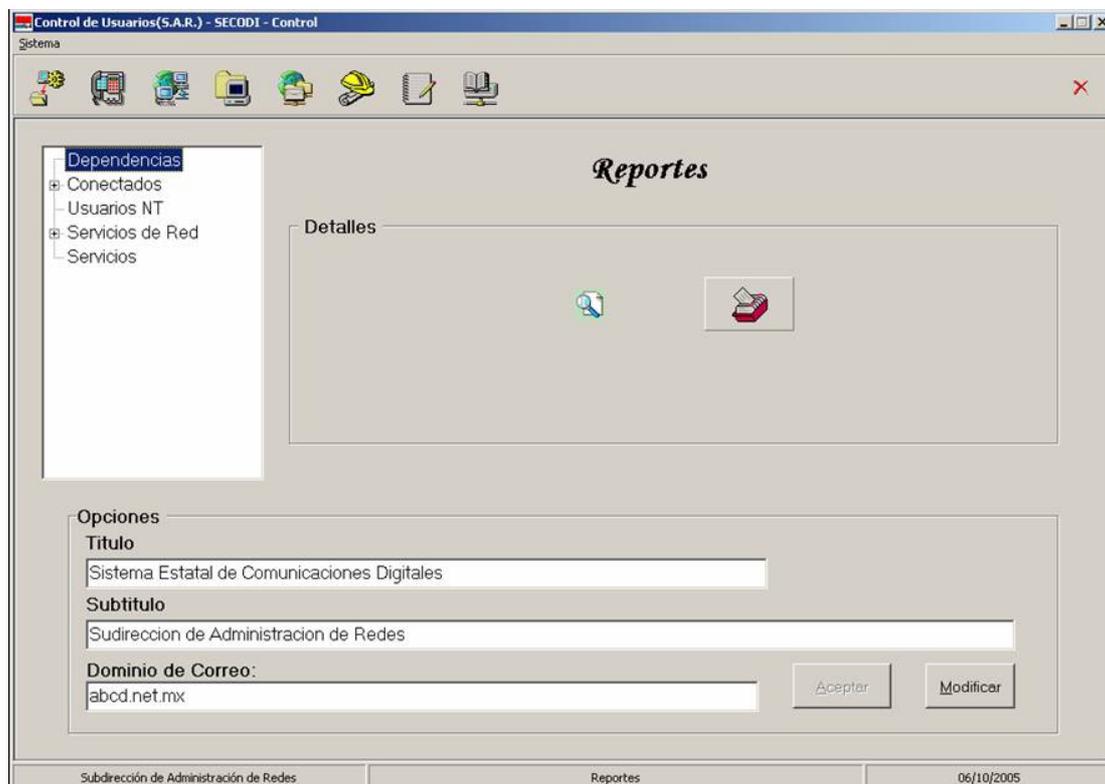


Figura 4.20 Reportes

En la figura 4.21 podemos observar la pantalla utilizada para llevar a cabo la actualización de los catálogos que se relacionan en los módulos del sistema.

La sección 20 del apéndice C nos muestra el código para este modulo.

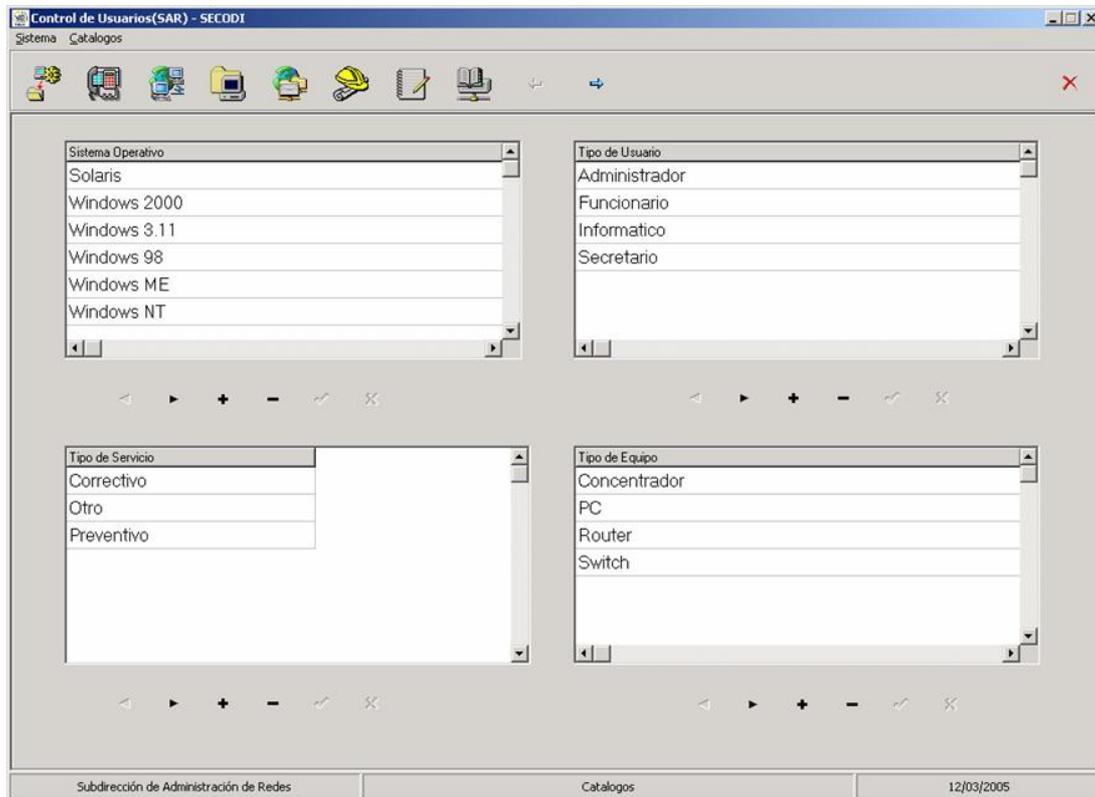


Figura 4.21 Alta de conceptos.

En esta pantalla podemos observar el módulo que contiene los catálogos o conceptos donde se permite capturar los tipos de Tarjetas de Red, de usuarios, servicios proporcionados, Equipo que se esta usando, sistemas operativos y las ubicaciones para las dependencias.

También se utilizan Cuestionarios para poder tener toda la información completa en el sistema, este procesos lo llevan a cabo las personas responsables del departamento de informática de cada una de las dependencias a los usuarios conectados a la red, para conocer esta información y sean registrados o actualizados los equipos existentes.

En los códigos mostrados en el apéndice C, y a los cuales se hace referencia en este capítulo, observamos un procedimiento denominado CAMBIO (); el cual se muestra en la sección 21 de dicho apéndice.

La figura 4.22 nos muestra el sitio WEB desde el cual se pueden hacer consultas, esta figura muestra los dos datos necesarios para ingresar al sitio que son: nombre de usuario y contraseña, estos datos son proporcionados en el módulo Dependencias, donde se dan de alta los responsables.

En la sección 24 del apéndice C, encontramos todo el código del CGI para validar los datos proporcionados por el usuario e ingresar al sitio.



Figura 4.22 Entrada al sitio WEB para consulta de datos.

La figura 4.23 nos muestra el sitio WEB desde el cual se pueden seleccionar cuatro opciones: Equipos Conectados, Usuarios de Correo, Usuarios de Internet y Solución a preguntas frecuentes, para ingresar a cualquiera de estas opciones, la seleccionamos y presionamos el botón *VER*. En la sección 24 del apéndice C, encontramos todo el código del CGI generar la pagina dinámica de la consulta.



Figura 4.23 Selección de opciones para la consulta Web

4.4 Salidas del Sistema

Las salidas que el sistema le presenta al usuario son muy variadas, desde un reporte en pantalla, reporte en papel, simple consulta, hasta la exportación de los datos a un documento Microsoft Word. A continuación observaremos algunos de los reportes y pantallas que genera el sistema como salida.

4.4.1 Reportes

En la figura 4.24 se visualiza la salida de un reporte por pantalla, el cual se puede enviar a la impresora para tenerlo en papel, también puede ser enviado ha archivo, este es generado con la herramienta Quick Report que viene incluida en el lenguaje de programación (Delphi 6). En la sección 18 del apéndice C podemos observar el código necesario para generar los reportes como el de la figura 4.24.

Nombre del responsable	Dependencia	Telefono	Rango IP
Pedro torres	dep 1	12421	192.168.0.7 192.168.0.8
Juan de Dios Meza Rodriguez	dep 2	12421214	192.168.0.5 192.168.0.6
Juan N asdasd	dep 3	45656656	194.168.0.3 192.168.0.4
Pedro Perez Pereyra	dep 4	124214214	192.168.0.1 192.168.0.2
Peter Norton	dep 5	12412421	192.168.10.2 192.168.10.110
			Total de dependencias: 5

Figura 4.24 Reporte de las Dependencias.

En la figura 4.25 se visualiza la salida de un reporte por pantalla (reporte de Dependencias), el es en este caso en una hoja de Microsoft Word. Este reporte se realiza con la ayuda de un control denominado ARWordReport, esto permite la interacción entre el lenguaje de programación Delphi y el programa Microsoft Word.

En la sección 19 del apéndice C podemos observar el código necesario para generar los reportes en Microsoft Word.

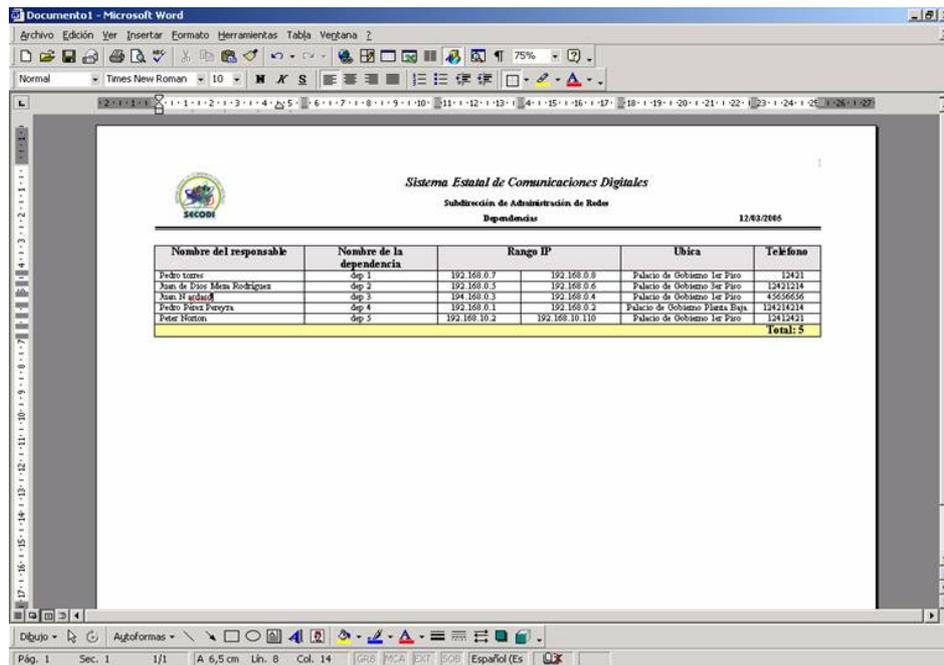


Figura 4.25 Reporte de Dependencias en una hoja de Word.

4.4.2 Reportes en el Web

En la figura 4.26, 4.27 y 4.28 se visualizan los reportes que obtenemos después de realizar una consulta desde la página Web, el cual se puede copiar y enviar a cualquier Procesador de palabras desde donde le podemos dar formato e imprimirlo.

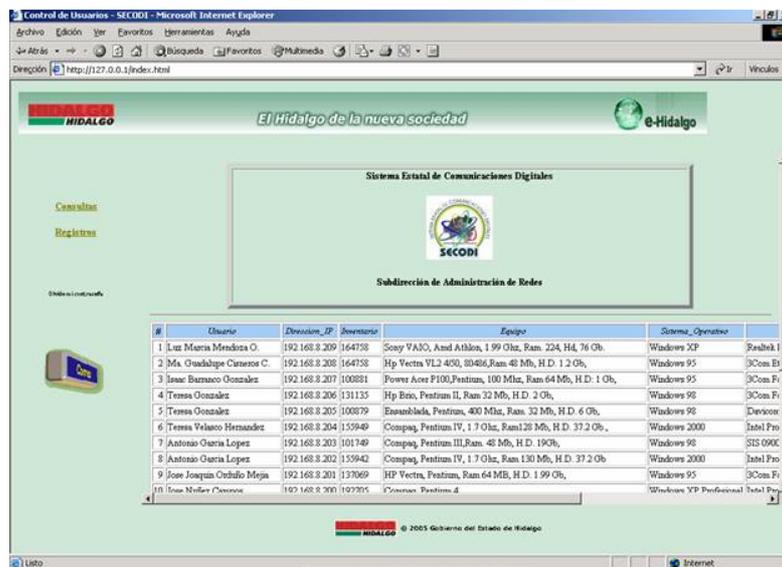


Figura 4.26 Reporte WEB de usuarios conectados



Figura 4.27 Reporte WEB de usuarios de correo electrónico



Figura 4.28 Solución a preguntas frecuentes

Capítulo 5

Implementación del Sistema

5.1 Determinación de Requerimientos

Para llevar a cabo la determinación de requerimientos se tomaron en cuenta los siguientes puntos:

- Se contaba con el equipo de cómputo necesario, el cual está a la vanguardia de la última tecnología, por lo que se estima un tiempo de vida al menos de 4 años.
- Se cuenta con el personal necesario para llevar a cabo las tareas que fuera necesario realizar, los cuales tienen el conocimiento suficiente para desempeñar todo lo que sea necesario.
- Las versiones de software con las que se cuenta son las últimas, en cuanto al lenguaje Visual y los manejadores de Base de Datos.
- No se implica ningún costo extra para la empresa. Ya que con el personal de sistemas con que se cuenta actualmente se cubren perfectamente las necesidades, tampoco se debe invertir en equipo de cómputo o nuevo software.

5.2 Requerimientos de Hardware y Software

En esta parte del documento se muestra todos los detalles de Hardware y Software necesarios para la instalación de S.A.R. Versión 1.0 tanto en cliente como en su parte de servidor.

1.- PLATAFORMA SQLSERVER-NT

Server

Windows NT 4.0 o superior

Windows Service Pack 3 mínimo

128 Mb. Memoria RAM mínimo

1 procesador Pentium o superior (deseable)

HD con controladoras independientes (el tamaño se definirá de acuerdo al volumen de datos)

MS SQL Server 7.0 Enterprise Edition (Service Pack 2) o superior (deseable versión 2000)

CLIENTE MULTIPLATAFORMA

Windows '98 o superior (deseable)

16 Mb RAM mínimo

10 Mb. Disco mínimo (considerando archivos necesarios para el BDE entre otros)

REQUERIMIENTOS QUE DEBE CUBRIR EL USUARIO PARA PODER EJECUTAR S.A.R.

Para que el usuario pueda utilizar S.A.R. no necesita tener grandes conocimientos en computación, ya que S.A.R. es muy accesible y fácil de utilizar.

ESPECIFICACIONES DE S.A.R. V 1.0

- Desarrollado para trabajar en arquitectura cliente-servidor.
- Comunicación mediante ODBC. Para la creación de un Origen de Datos ver el apéndice A donde se muestran detalladamente cada uno de los pasos.
- Flujo a través de la red de paquetes TCP/IP
- Creación y acceso de usuarios controlado por el DBA
- Asignación de niveles de seguridad a usuarios desde la aplicación

5.3 Instalación del Sistema

La instalación del Sistema de Administración de Red(S.A.R.) se divide en tres partes, la parte del cliente, la parte del servidor y la pagina Web para las consultas de los usuarios, a continuación observaremos la instalación para ambas partes de manera mas detallada.

5.3.1 Instalación cliente

La instalación se realiza desde un archivo llamado Install.exe el cual esta dentro del primer disco de instalación.

La figura 5.1 nos muestra la pantalla de inicio para la instalación.



Figura 5.1 Inicio de la instalación

A continuación en la figura 5.2 se muestra la pantalla de bienvenida a la instalación.



Figura 5.2 Pantalla de bienvenida a la instalación

En la figura 5.3 se puede observar que la aplicación genera el directorio por default donde se copiarán los archivos necesarios para la ejecución del S.A.R.. La ruta puede ser modificada por el usuario, pero se recomienda que se respete el directorio default para evitar alguna anomalía al momento de la ejecución del Software.

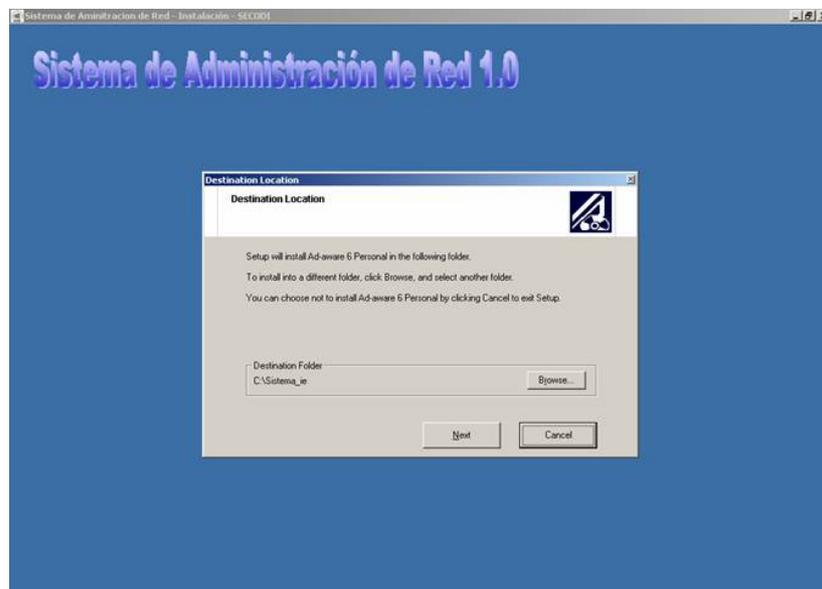


Figura 5.3 Pantalla de elección de directorio para la instalación

Una vez seleccionada la ruta de instalación, debemos elegir la opción *Instalación para cliente* como lo muestra la figura 5.4, para que esta se encargue de la instalación de los archivos correspondientes para la ejecución del S.A.R. 1.0.



Figura 5.4 Pantalla de opciones para la instalación

Una vez que se ha elegido la carpeta así como el tipo de instalación el programa comienza la instalación como se puede ver en la figura 5.5.

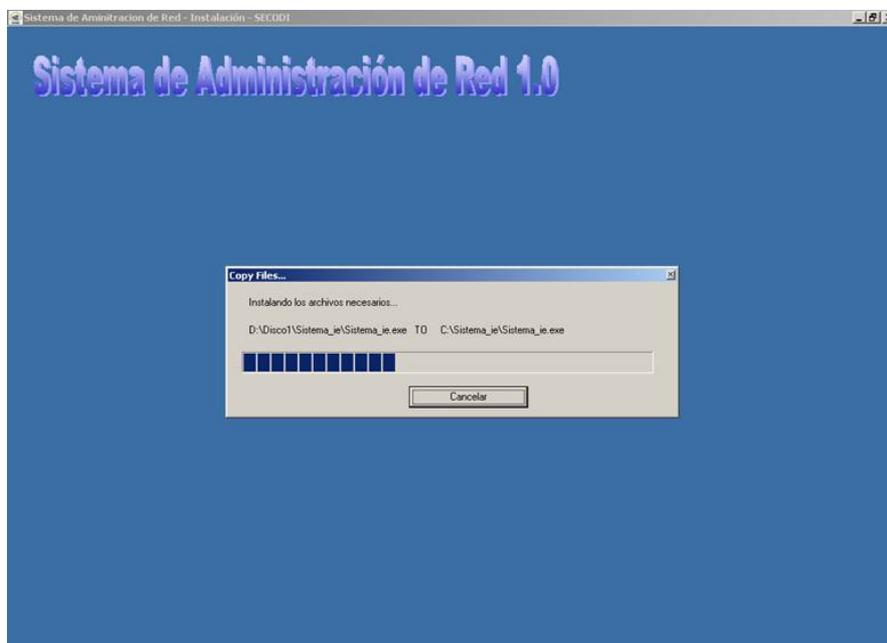


Figura 5.5 Proceso de instalación

5.1.2 Instalación Servidor

Para la instalación del servidor es necesario tener instalado previamente el servidor para Bases de Datos (MICROSOFT SQL SERVER). La instalación se realiza desde un archivo llamado Install.exe el cual esta dentro del primer disco de instalación.

Para esta instalación debemos llegar a la elección del tipo de instalación, como lo muestra la figura 5.4 en la cual seleccionaremos la opción *Instalación de Servidor* la cual se encargara de realizar la copia de archivos necesarios y la creación de una Base de Datos vacía.

Si existiera alguna falla en la instalación de la Base de Datos por cuestiones del sistema, se puede realizar de manera manual desde el *Query Analyzer*, ya que se incluye un archivo llamado *crea.sql* (ver apéndice C, sección 1) en cual realiza esta tarea. La figura (5.6) siguiente nos muestra la ventana del Query Analyzer.

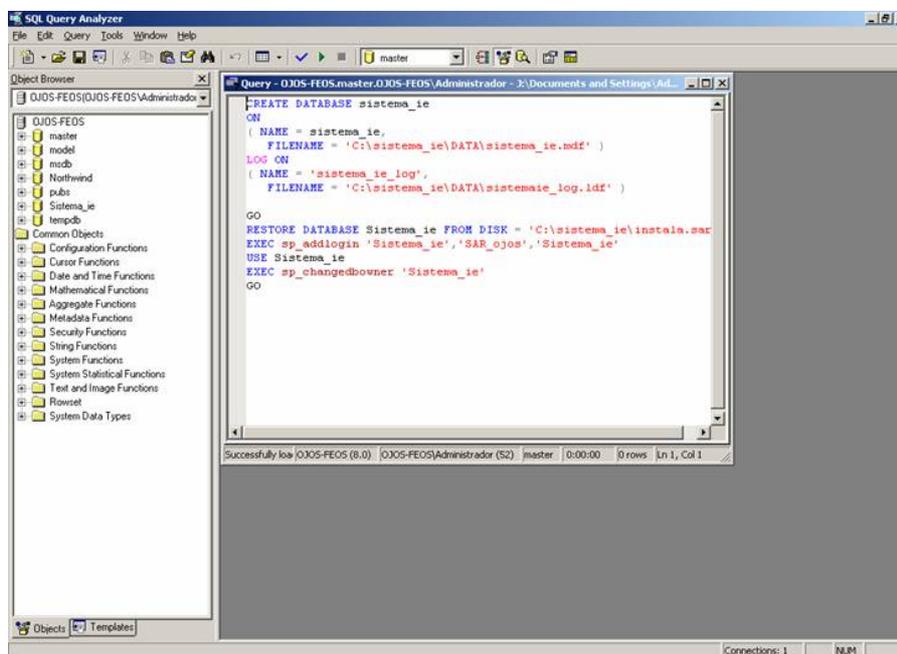


Figura 5.6 Instalación de la Base de Datos desde el Query Analyzer.

5.1.3 Instalación de la Página Web.

Para la instalación de la Pagina Web es necesario tener instalado previamente la herramienta IIS (Internet Information Server, ver Apéndice B).

Para esta instalación debemos llegar a la elección del tipo de instalación, como lo muestra la figura 5.4 en la cual seleccionaremos la opción *Instalación de la Pagina Web* la cual se encargara de realizar la copia de archivos necesarios a los directorios correspondientes para la ejecución de la Pagina Web desde el Navegador.

La pagina por default es *Index.html* la cual se muestra en la figura 5.7. Para la generación del CGI que crea la pagina Web el código lo podemos ver en el apéndice C, sección 24; este CGI es programado en Delphi versión 6.0, donde se utiliza código HTML que es usado para la generación de los formularios con las opciones para la consulta desde el sitio Web, en la sección 25 del mismo apéndice se encuentra el código HTML.



Figura 5.7 Pagina Web Default

5.2 Pruebas y Resultados

Se llevan a cabo dos tipos de pruebas para ir eliminando los errores o detalles que van surgiendo al momento del desarrollo del software.

La primera parte de las pruebas se realizan mientras se va desarrollando el programa, es decir, al momento de ir realizando la programación se van considerando las posibles fallas que puedan surgir, de tal manera que al finalizar el desarrollo sea confiable y de este modo evitar las modificaciones extensas a nivel de programación.

La prueba consiste en llenar con datos de prueba ya sea validos o inválidos para saber si el programa generara algún error al momento de guardar o hacer uso en alguna consulta de dichos datos.

La segunda parte de estas pruebas las realizan los administradores una vez que se ha terminado el desarrollo completo donde ellos se encargan de la introducción de datos reales y generación de consultas para verificar que toda la información que se pide es regresada correctamente.

Si esta segunda prueba falla se deben reprogramar algunas secciones del S.A.R. para que pueda cumplir con los requerimientos necesarios para un óptimo desempeño.

5.3 Especificaciones del Modo de Uso

Los iconos y teclas de función usados dentro del S.A.R. tienen una imagen con relación a la acción que realizan además de una leyenda que se muestra al colocar el puntero del Mouse sobre ellos.

5.3.1 Botones de Acción

Los botones de acción que se presentan a continuación se encuentran situados en la parte superior de la ventana principal la cual permite un acceso más rápido a cada una de las funciones y de este modo le es de gran utilidad a los usuarios realizar las acciones correspondientes de acuerdo al uso que le este dando al S.A.R..

Catálogos



Este botón nos envía a la ventana donde se agregan todas las opciones que se utilizarán al momento de dar de alta equipos, por ejemplo, los tipos de tarjetas de Red. Esta opción cuenta con dos páginas debido a que es un poco extensa la información que aquí se encuentra. Ver código para este módulo en la sección 20 del apéndice C.

Dependencias



En esta sección damos de alta todas las dependencias que forman parte del Gobierno del Estado. Aquí se asignan las cuentas de usuario y pasaporte que serán usadas en la página Web para consultas de los equipos conectados de cada Dependencia. Ver código para este módulo en la sección 5 del apéndice C.

Equipo Conectados



En este módulo se dan de alta todos los equipos que se encuentran conectados a la red del Gobierno del Estado, desde una computadora hasta un Router. Ver código para este módulo en la sección 2 del apéndice C.

Usuarios NT



Esta ficha está diseñada especialmente para todos los usuarios del Dominio de la Dirección General de Modernización y Sistemas en la cual aparecen todos los nombres de usuario y pasaporte de las cuentas. Ver código para este módulo en la sección 7 del apéndice C.

Usuarios de Correo



Por medio de este botón se agregan usuarios que tienen asignadas cuentas de correo electrónico dentro del dominio hidalgo.gob.mx. Ver código para este módulo en la sección 9 del apéndice C.

Servicios



Este icono nos envía a una bitácora de servicios los cuales se van realizando todos los días en donde se agregan datos como la falla que se reporta, la solución, la Dependencia a la que se le realizó el servicio, etc. Ver código para este módulo en la sección 10 del apéndice C.

Reportes



Este botón es el más importante dentro del S.A.R. ya que mediante este se realizan las consultas necesarias para obtener los reportes de acuerdo a las necesidades del usuario. Los reportes pueden ser generados en pantalla como una simple consulta además de que se puede imprimir o enviarlos directamente a un archivo de Microsoft Word con un formato previamente diseñado para ser entregados a las dependencias o a la persona que necesite esta información. Ver código para este módulo en la sección 14 del apéndice C.

Preguntas Frecuentes



Como ya es común en las páginas Web existe una sección de preguntas frecuentes y mediante este botón se actualiza directamente esa sección, no es necesario modificar directamente el sitio, es decir, la página Web donde está la sección, esto hace más fácil de mantener al día este tipo de tips o dudas a los usuarios. Ver código para este módulo en la sección 13 del apéndice C.

Atrás



Este botón sirve para cambiar a la página anterior dentro de la sección de Catálogos.

Siguiente



Este botón sirve para cambiar a la página siguiente dentro de la sección de Catálogos.

Salir



Termina con la ejecución del S.A.R.

Todos los códigos señalados en la parte superior hacen uso de un procedimiento auxiliar para la verificar el estado de conexión de las tablas el cual está en la sección 21 del apéndice C.

5.3.2 Teclas de función

Las teclas de función permiten un acceso más rápido a cada una de las funciones y de este modo le es de gran utilidad a los usuarios realizar las acciones correspondientes de acuerdo a las necesidades de uso. La figura 5.8 nos muestra el menú completo en donde aparecen todas las teclas de función que contiene el programa, la descripción se realizó en la sección anterior.



Figura 5.8 Barra de menú

5.4 Utilerías S.A.R.

Las utilerías S.A.R. es un anexo del sistema que se divide en tres secciones, Copia de seguridad de la Base de datos, Restauración de la Base de Datos e instalación de la Base de Datos.



Figura 5.9 Pantalla de bienvenida al S.A.R. Utilerías

La pantalla principal del módulo de Utilerías es la que se muestra en la figura 5.10

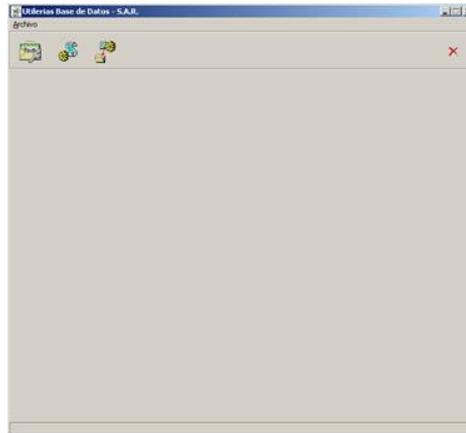


Figura 5.10 Ventana principal de Utilerías

5.4.1 Copia de seguridad

La sección de copia de seguridad Genera una copia de toda la base de datos en la ruta y nombre que el usuario indique, el archivo que se genera tiene por default la extensión sar para una fácil identificación de que tipo de archivo es. El código para la realización de este proceso lo podemos ver en el apéndice C sección 22.

La copia de seguridad realiza la tarea que en el Administrador Corporativo de SQL Server se llama BACKUP, para una información mas detallada el Apéndice C nos muestra como se lleva a cabo este proceso. La ventana de este modulo es que a continuación se muestra en la figura 5.11.

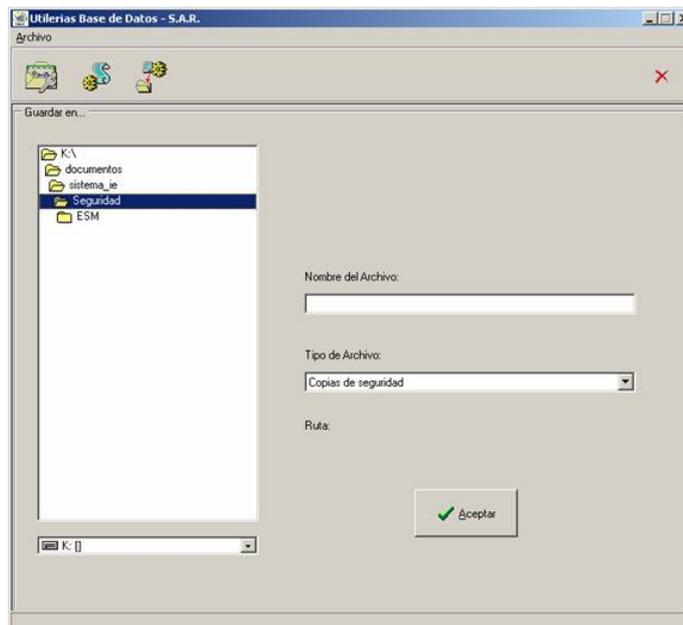


Figura 5.11 Pantalla para realizar copias de Seguridad

El código del botón Aceptar lo podemos ver detalladamente en el Apéndice C, sección 23.

5.4.2 Restauración

La parte restauración se encarga de abrir algún archivo que se ha realizado mediante la copia de seguridad en la ruta y nombre que el usuario indique, el archivo que se indica debe tener por default la extensión sar para que este pueda ser leído por el software.

La restauración realiza la tarea que en el Administrador Corporativo de SQL Server que se llama RESTORE, para una información mas detallada, el Apéndice C nos muestra como se lleva a cabo este proceso.

La ventana de este modulo es que a continuación se muestra en la figura 5.12.

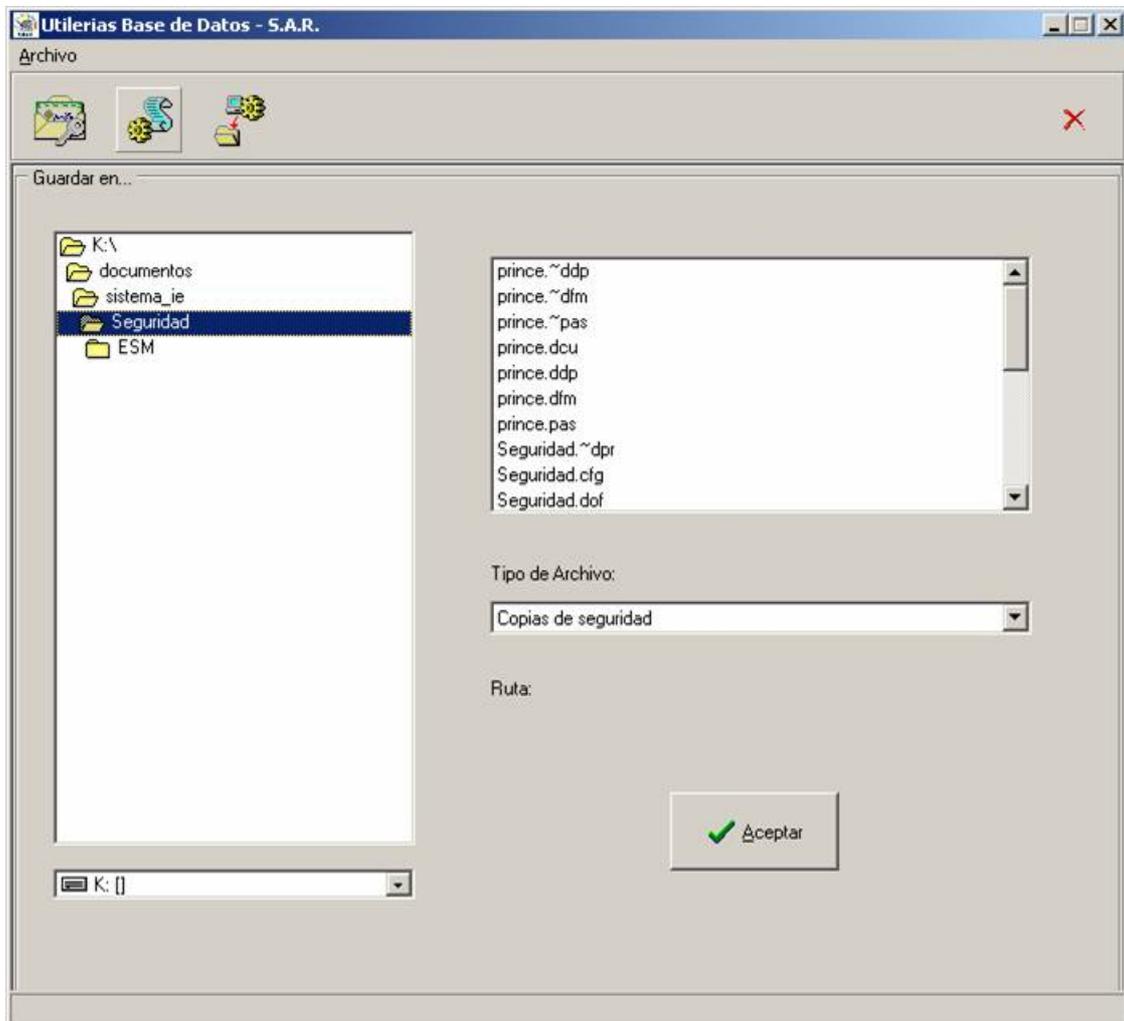


Figura 5.12 Pantalla para realizar la restauración de la Base de Datos

El código del botón Aceptar lo podemos ver detalladamente en el Apéndice C, sección 23.

5.4.3 Instalación de la Base de Datos

Este modulo tiene una función importante, ya que en caso de alguna anomalía o alguna perdida de información o que nuestro servidor de Bases de Datos sufra un daño, no es necesario instalar nuevamente todo el S.A.R. completo, esta herramienta se encarga solamente de la instalación de la Base de Datos vacía, la cual posteriormente se puede restaurar con la herramienta de *Restauración* contenida en Software de Utilerías y con ello poder nuevamente tener toda la información hasta la ultima copia de seguridad que el usuario realice. Se recomienda al usuario que haga una copia de seguridad todos los días.

Para poder realizar este proceso se debe escribir un nombre de usuario y contraseña ya que es una herramienta un poco delicada para su uso y puede ocasionar perdida de datos importantes si no se usa adecuadamente. La figura 5.13 muestra el cuadro de dialogo de confirmación para llevar a cabo esta operación.



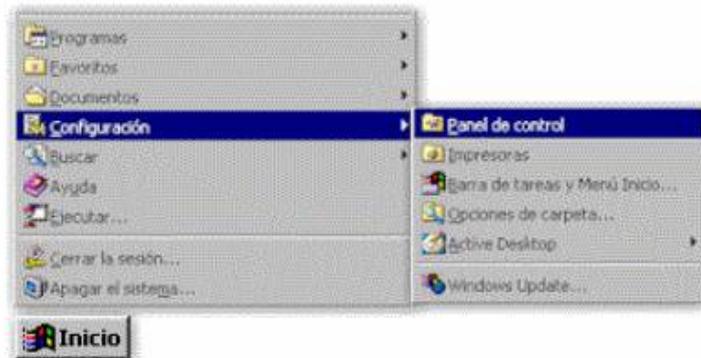
Figura 5.13 Cuadro de dialogo de confirmación.

El código que realiza esta operación se encuentra en el apéndice C sección 22.

Apéndice A

Sección 1. Crear Origen de Datos ODBC

Primero vamos a buscar a ODBC, que está en el Panel de Control.



Las aplicaciones creadas específicamente para Windows 95, 98 y NT usan el ODBC de 32 bits; pero algunos sistemas conservan un ODBC de 16 bits para las aplicaciones de legado que corrieran o corren en Windows 3.11.

El [Data Source Administrator](#) del [Open DataBase Connectivity](#), u ODBC. Lo que sigue es crear una fuente u origen de datos, pero antes unas explicaciones:

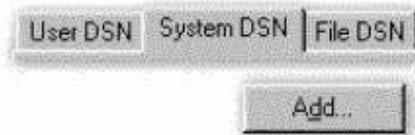


Vas a notar que las primeras tres solapas se refieren a [User DSN](#), [System DSN](#) y [File DSN](#).

- **User DSN, nombre del origen de datos para el usuario.** A veces, una máquina es utilizada por más de un usuario, los orígenes de datos declarados aquí son exclusivos del usuario.
- **System DSN, nombre del origen de datos para el sistema.** Todos los usuarios de la máquina tienen acceso a estos orígenes de datos.
- **User DSN, nombre del origen de datos en archivo.** Se crea un archivo con la extensión DSN, que sirve como origen de datos y puede ser distribuido a otros usuarios. Este origen es el que usa Excel por omisión cuando hace consultas, cuidado con eso.

Está otra solapa importante que es **ODBC Drivers** o **Controladores ODBC**. Aquí se ven todos los controladores disponibles en la máquina. De esta forma puedes consultar si dispones del controlador que necesitas y si es la versión conveniente. Regularmente los controladores de bases de datos vienen con un programa **SETUP** que los instala y quedan dados de alta en esta lista.

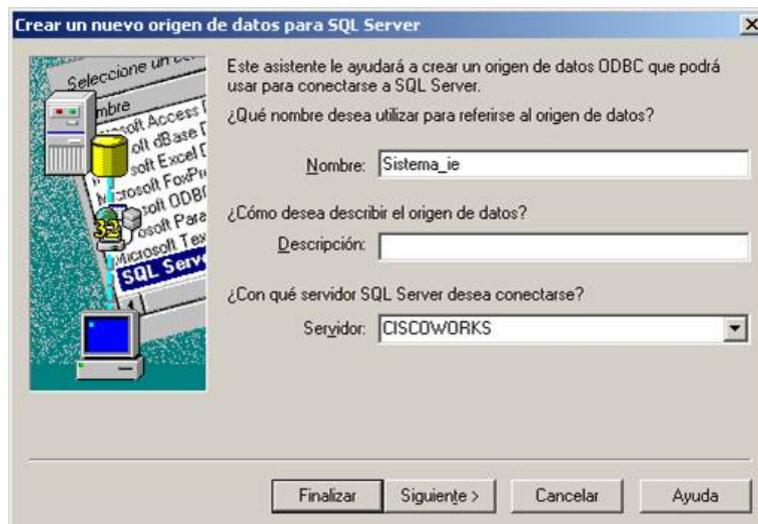
Seleccionar la solapa de DSN, nombre de origen de datos, que mejor se ajuste a mis requerimientos. S.A.R. no trabaja con archivos DSN, pero sí con orígenes de usuario y de sistema, así que, seleccionamos la solapa System DSN, nombre de origen de datos del sistema y presionamos el botón Add..., agregar....



Luego señalamos el controlador o driver del tipo de base de datos que queremos acceder: **SQL Server** y presionamos Finalizar; pero fíjate que todavía no acabamos, eso de Finalizar es algo que... ni hablar.



Lo que tenemos que hacer ahora bautizar al nuevo origen de datos con un nombre peculiar y distintivo, este puede ser un nombre relacionado con la base de datos que se esta enlazando, además se debe seleccionar el servidor hacia el cual nos queremos conectar.



Ahora debemos indicar el nombre de usuario y contraseña en el administrador de ODBC. En la sección Configuración del cliente encontraremos opciones como las Canalizaciones por nombre, conexión TCP/IP entre otras, dependiendo de la red y otros factores elegiremos el más adecuado.



Una vez que se ha completado de acuerdo al nombre de usuario aparece seleccionada por omisión la base de datos.





Una vez que hemos terminado de la configuración adicional de idioma, entre otras opciones seleccionamos el botón finalizar el cual nos mostrara una ventana informativa de todas las opciones que hemos seleccionado durante el proceso de creación del nuevo Origen de Datos. En esta ventana aparece un botón denominado *Probar origen de datos* el cual realiza una prueba de conexión para verificar que el nuevo origen de datos funciona adecuadamente. Las siguientes dos figuras muestran estas ventanas.



Sección 2. Crear una cadena de conexión ADO

El primer paso en la creación de una aplicación de datos en Web consiste en proporcionar un método para que ADO encuentre e identifique el origen de datos. Para ello se utiliza una cadena de conexión, una serie de argumentos separados mediante un punto y coma que definen parámetros como el proveedor del origen de datos y la ubicación del mismo. ADO utiliza la cadena de conexión para identificar el proveedor OLE DB y para dirigir al proveedor al origen de datos. El proveedor es un componente que representa el origen de datos y que expone la información en la aplicación en forma de conjuntos de filas.

En la tabla siguiente se enumeran las cadenas de conexión de OLE DB para varios orígenes de datos habituales:

Origen de datos	Cadena de conexión OLE DB
Microsoft(r) Access	Provider=Microsoft.Jet.OLEDB.4.0;Data Source= ruta física de acceso al archivo .mdb
Microsoft SQL Server	Provider=SQLOLEDB.1;Data Source= ruta de acceso a la base de datos del servidor
Oracle	Provider=MSDAORA.1;Data Source= ruta de acceso a la base de datos del servidor
Microsoft Indexing Service	Provider=MSIDXS.1;Data Source= ruta de acceso al archivo

Para proporcionar compatibilidad con versiones anteriores, el proveedor OLE DB para ODBC admite la sintaxis de las cadenas de conexión ODBC. En la tabla siguiente se enumeran las cadenas de conexión ODBC que se utilizan habitualmente:

Controlador del origen de datos	Cadena de conexión ODBC
Microsoft Access	Driver={ Microsoft Access Driver (*.mdb)};DBQ= ruta física de acceso al archivo .mdb
SQL Server	DRIVER={ SQL Server };SERVER= ruta de acceso al servidor
Oracle	DRIVER={Microsoft ODBC for Oracle };SERVER= ruta de acceso al servidor
Microsoft Excel	Driver={ Microsoft Excel Driver (*.xls)};DBQ= ruta física de acceso al archivo .xls ; DriverID=278
Microsoft Excel 97	Driver={ Microsoft Excel Driver (*.xls)};DBQ= ruta física de acceso al archivo .xls ;DriverID=790
Paradox	Driver={Microsoft Paradox Driver (*.db)};DBQ= ruta física de acceso al archivo .db ;DriverID=26
Texto	Driver={Microsoft Text Driver (*.txt;*.csv)};DefaultDir= ruta física de acceso al archivo .txt
Microsoft Visual FoxPro (r)	Driver={Microsoft Visual FoxPro

(con un contenedor de bases de datos)	Driver};SourceType=DBC;SourceDb= ruta física de acceso al archivo .dbc
Microsoft Visual FoxPro (sin un contenedor de bases de datos)	Driver={Microsoft Visual FoxPro Driver};SourceType=DBF;SourceDb= ruta física de acceso al archivo .dbf

Conectarse al origen de datos

ADO proporciona el objeto `Connection` para establecer y administrar las conexiones entre las aplicaciones y los orígenes de datos compatibles con OLE DB o las bases de datos compatibles con ODBC. El objeto `Connection` incorpora propiedades y métodos que se pueden utilizar para abrir y cerrar conexiones con bases de datos, y para enviar consultas de actualización de la información.

Para establecer una conexión con una base de datos, cree primero una instancia del objeto `Connection`. Por ejemplo, la secuencia de comandos siguiente crea una instancia del objeto `Connection` y procede a abrir una conexión:

```
<% 'Crea un objeto Connection. Set cn =
Server.CreateObject("ADODB.Connection") 'Abre una conexión mediante la cadena
de conexión OLE DB. cnn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:DatosMercadoVentasPrevistas.mdb" %>
```

La cadena de conexión no contiene espacios en blanco ni antes ni después del signo igual (=). En este caso, el método `Open` del objeto `Connection` se refiere a la cadena de conexión.

Ejecutar consultas SQL con el objeto Connection

Con el método `Execute` del objeto `Connection` puede emitir comandos al origen de datos, como consultas de SQL (Lenguaje de consulta estructurado). (SQL, lenguaje estándar para comunicarse con bases de datos, define comandos para recuperar y actualizar información.) El método `Execute` acepta parámetros que especifiquen el comando (o la consulta), el número de registros de datos a los que afecta y el tipo de comando que se utiliza.

La siguiente secuencia de comandos utiliza el método `Execute` para enviar una consulta con un comando `INSERT` de SQL, que inserta datos en una tabla concreta de la base de datos. En este caso, el bloque de la secuencia de comandos inserta el nombre José Lugo en una tabla de la base de datos llamada `Customers`.

```
<% 'Define la cadena de conexión OLE DB. strConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:DatosEmpleados.mdb" 'Crea la
instancia del objeto Connection y abre una conexión con la base de datos. Set cn =
```

```
Server.CreateObject("ADODB.Connection") cnn.Open strConnectionString 'Define la
instrucción SELECT de SQL. strSQL = "INSERT INTO Customers (FirstName, LastName)
VALUES ('José','Lugo)'" Utiliza el método Execute para enviar una consulta SQL a la
base de datos. cnn.Execute strSQL,,adCmdText + adExecuteNoRecords %>
```

Observe que se especifican dos parámetros en la instrucción que se utiliza para ejecutar la consulta: `adCmdText` y `adExecuteNoRecords`. El parámetro opcional `adCmdText` especifica el tipo de comando e indica que el proveedor debe evaluar la instrucción de consulta (en este caso, una consulta SQL) como una definición textual de un comando. El parámetro `adExecuteNoRecords` indica a ADO que no debe crear un conjunto de registros de datos si no se devuelven resultados a la aplicación. Este parámetro sólo funciona con los tipos de comandos definidos como texto, como las consultas SQL, o con procedimientos almacenados de bases de datos. Aunque los parámetros `adCmdText` y `adExecuteNoRecords` son opcionales, debe especificarlos al utilizar el método `Execute` para mejorar así el rendimiento de la aplicación de datos.

Además del comando `INSERT` de SQL, puede utilizar los comandos `UPDATE` y `DELETE` de SQL para modificar y quitar información de la base de datos.

Con el comando `UPDATE` de SQL puede modificar los valores de los elementos de una tabla de la base de datos. La siguiente secuencia de comandos usa el comando `UPDATE` para cambiar todos los campos `FirstName` de la tabla `Customers` a Juan en todas las filas cuyo campo `LastName` contenga el apellido Soto.

```
<% Set cn = Server.CreateObject("ADODB.Connection") cnn.Open
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:DatosEmpleados.mdb" cn.Execute
"UPDATE Customers SET FirstName = 'Juan' WHERE " & _ & "LastName = 'Soto'
",,adCmdText + adExecuteNoRecords %>
```

Para quitar determinados registros de una tabla de la base de datos, utilice el comando `DELETE` de SQL. La siguiente secuencia de comandos quita todas las filas de la tabla `Customers` cuyo apellido sea Soto:

```
<% Set cn = Server.CreateObject("ADODB.Connection") cnn.Open
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:DatosEmpleados.mdb" cn.Execute
"DELETE FROM Customers WHERE LastName = 'Soto'",,adCmdText +
adExecuteNoRecords %>
```

Apéndice B

Instalación de IIS en Windows 2000

Como cualquier otro software de Windows, la instalación de IIS 5.0 es tan sencilla como hacer un doble clic de ratón. Normalmente es uno de los componentes de Windows 2000 que viene seleccionado por defecto. Si no fuera así, en la propia instalación de Windows 2000 se puede seleccionar bajo el epígrafe Componentes de Windows.

Si se desea instalar manualmente habría que ir al Panel de Control->Añadir o Quitar Programas y hacer clic sobre el icono Agregar o Quitar componentes de Windows. Una vez lanzado el Asistente para Componentes de Windows, seleccionar de la lista Servicios de Internet Information Server.

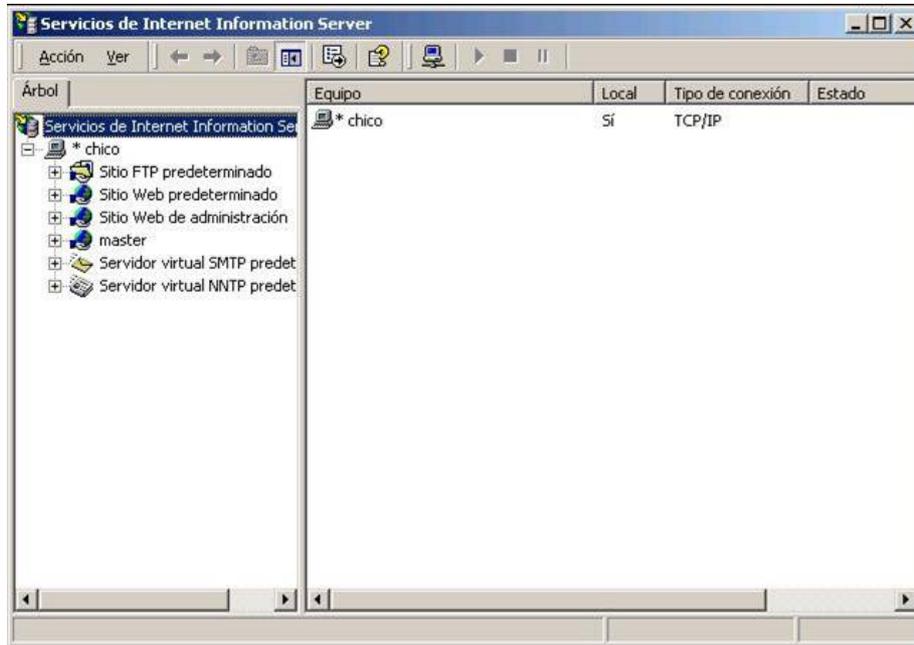


Asistente para componentes de Windows

ADMINISTRACIÓN DE SITIOS WEB

En este punto, nos centraremos en las tareas de administración del servidor Web y Ftp de IIS, aunque IIS puede realizar las funciones de servidor SMTP (Send Mail Transfer Protocol) y de servidor NNTP (o servidor de noticias).

La herramienta recomendable de administración del software IIS será el snap-in de la MMC (Microsoft Management Console) o Administrador de servicios de Internet.



Administrador de servicios de Internet snap-in

Creación de un sitio Web

Internet Information Server incluye un sitio web, un sitio FTP, un sitio SMTP y un sitio NNTP configurados por defecto. Esto no significa que deba limitarse a un único sitio; se pueden crear sitios virtuales en el mismo equipo

Los sitios Web están almacenados en directorios según una estructura lógica. Existen dos tipos de directorios: directorios principales y directorios virtuales.

- Directorio principal: En el caso del sitio web predeterminado suele ser el directorio `c:\inetpub\wwwroot`. En este directorio colgarán nuestras páginas web.
- Directorio virtual: se utiliza cuando el sitio web está distribuido entre varios directorios, unidades o equipos.

El primer paso para crear varios sitios web en su servidor consistirá en configurar los directorios principales predeterminados. Estos directorios pueden residir en el disco local o en una unidad de red. Bastará utilizar el explorador de Windows para crear una nueva carpeta.

A continuación, se iniciará el Administrador de servicios de Internet. En el menú Acción, seleccione Nuevo sitio web, para iniciar el Asistente para crear un sitio web. Haga clic en Siguiente para pasar a la pantalla de introducción de datos. El primer cuadro de diálogo nos pedirá una descripción del sitio, la cual lo identificará en la MMC.



The screenshot shows a dialog box titled "Asistente para crear un sitio Web" with a close button (X) in the top right corner. The main heading is "Descripción del sitio Web" and the instruction reads: "Escriba la descripción del sitio Web para que los administradores puedan identificarlo." Below this, there is a text input field labeled "Descripción:" containing the text "master". At the bottom of the dialog, there are three buttons: "< Atrás", "Siguiente >", and "Cancelar".

Descripción del sitio web

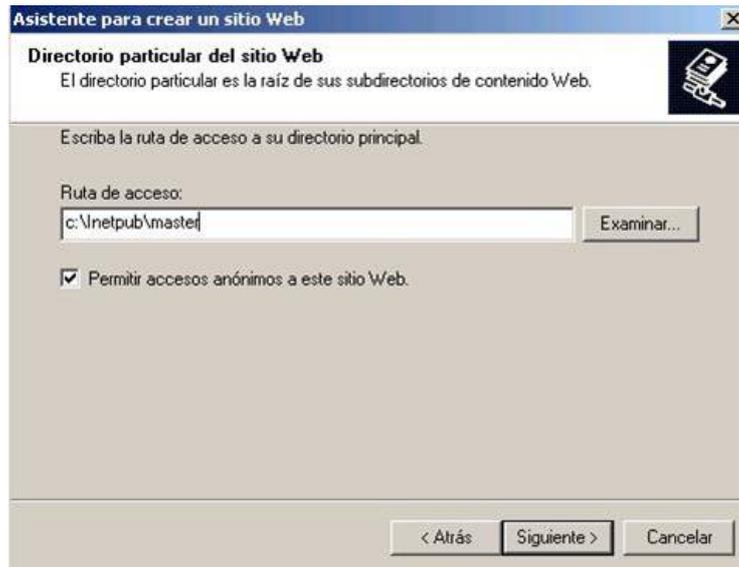
A continuación, aparecerá el cuadro de diálogo Dirección IP y configuración de puerto. En esta sección, también podemos definir el nombre de encabezado de host que nos permitirá crear un sitio virtual. Eso si, habrá que tener en cuenta que tendremos que añadir esa información a un servidor DNS (normalmente con una directiva CNAME) o en nuestro caso añadiendo la correspondiente entrada al fichero hosts.



The screenshot shows a dialog box titled "Asistente para crear un sitio Web" with a close button (X) in the top right corner. The main heading is "Dirección IP y configuración de puerto" and the instruction reads: "Especifique la dirección IP y la configuración de puerto para el nuevo sitio Web." Below this, there are several fields: "Escriba la dirección IP para este sitio Web:" with a dropdown menu showing "158.42.49.62"; "Puerto TCP que debería usar este sitio Web: (predeterminado: 80)" with a text input field containing "8080"; "Encabezado del host para este sitio: (predeterminado: ninguno)" with a text input field containing "master"; and "Puerto SSL para este sitio Web (predeterminado: 443)" with an empty text input field. At the bottom, there is a note: "Para obtener más información consulte la documentación de IIS." and three buttons: "< Atrás", "Siguiente >", and "Cancelar".

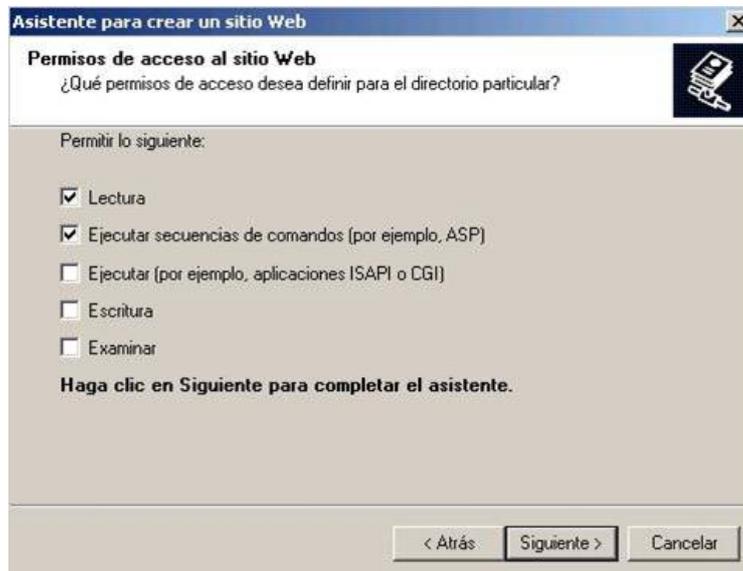
Dirección IP y configuración del puerto

El siguiente paso es definir el Directorio particular, donde escribiremos la ruta de acceso a la carpeta que hemos creado anteriormente. Si queremos que todo el mundo (sin autenticación previa) acceda a nuestro sitio web, dejaremos definido el ítem Permitir accesos anónimos a este sitio Web.



Directorio particular

Una vez definido el Directorio particular, aparecerá el cuadro de diálogo Permisos de acceso al sitio web.



Permisos de acceso al sitio web

Esta pestaña nos permitirá definir los permisos adecuados para que los clientes tengan acceso a este sitio web:

- Lectura. Permite que los clientes vean páginas de este sitio.

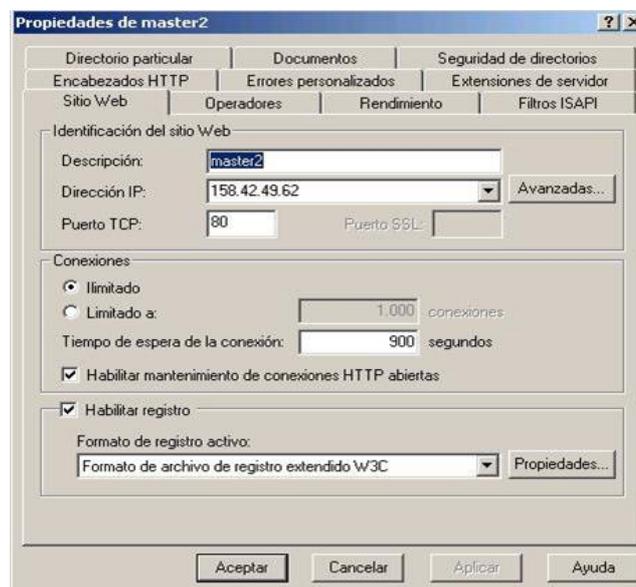
- Ejecutar secuencias de comandos. Permite que los clientes soliciten páginas con código ASP y que se ejecute dicho código.
- Ejecutar. Esta opción permite la ejecución de aplicaciones CGI o ISAPI en este sitio.
- Escritura. Si se activa esta opción, los clientes podrán cargar, eliminar o transferir archivos a este directorio.
- Examinar. Permite que los clientes examinen el contenido de los directorios

Si hemos seguido los pasos anteriores en MMC tendremos una nueva entrada cuyo nombre se corresponderá con la descripción del sitio y por tanto habremos creado un nuevo sitio web.

Configuración de un sitio Web

Cada sitio web tiene asociadas una serie de propiedades que definen su comportamiento. Por tanto, el administrador es libre de cambiar este comportamiento modificando sus propiedades. Estas propiedades se pueden modificar a través de las páginas de propiedades, y pueden referirse al sitio, al directorio o a un fichero en cuestión.

La página de propiedades de un sitio web se obtiene en la MMC pulsando el botón derecho sobre el sitio web anteriormente definido y eligiendo el menú propiedades.



Propiedades Sitio web

Muchas de las propiedades que podemos definir aquí ya las hemos visto a la hora de definir el nuevo sitio web. Nos centraremos en este momento en aquellas que creemos son importantes para un buen funcionamiento del servidor.

- **Sitio Web.** En esta lengüeta, además de definir la identificación del sitio web, podemos definir el número de conexiones que aceptará nuestro servidor web. En el caso de estar ejecutando IIS sobre Windows 2000 Profesional, existe una limitación de 10 conexiones. También podremos habilitar un registro o log de los accesos y errores del sitio web.
- **Operadores.** Los operadores del sitio web son usuarios definidos en Windows 2000 que poseen permisos para alterar la configuración y el funcionamiento del servidor Web. Aquí añadiremos aquellos usuarios que deseamos administren el sitio web.
- **Rendimiento.** En esta pestaña podremos ajustar una serie de parámetros que influirán en el rendimiento del sitio web. Los parámetros que se configuran para cada sitio, prevalecen sobre los definidos en el servidor
- **Documentos.** Aquí definiremos el documento predeterminado que se mostrará si se invoca este sitio directamente sin indicar una página concreta.
- **Encabezados HTTP.** Utilizaremos esta pestaña para configurar los valores que se enviarán al navegador en el encabezado de la página HTML.

Directorios Virtuales

Los directorios virtuales son directorios lógicos, que pertenecerán a la estructura de directorios que puede percibir el usuario que se conecta a nuestro servidor, pero que se corresponde con directorios físicos que se encuentran en ubicaciones distintas del directorio principal del servidor.

Los directorios virtuales se crean definiendo un alias que hace referencia a un directorio físico, de forma que cuando se navega por el servidor web el usuario verá dicho directorio como si fuese un directorio que cuelga directamente del directorio principal del servidor.

Para crear un directorio virtual, seleccionaremos el sitio web deseado y con el botón derecho elegiremos Nuevo y a continuación Directorio Virtual, para lanzar el asistente de creación de directorios virtuales. Lo primero que nos solicitará será el nombre del alias que le queremos dar al directorio.

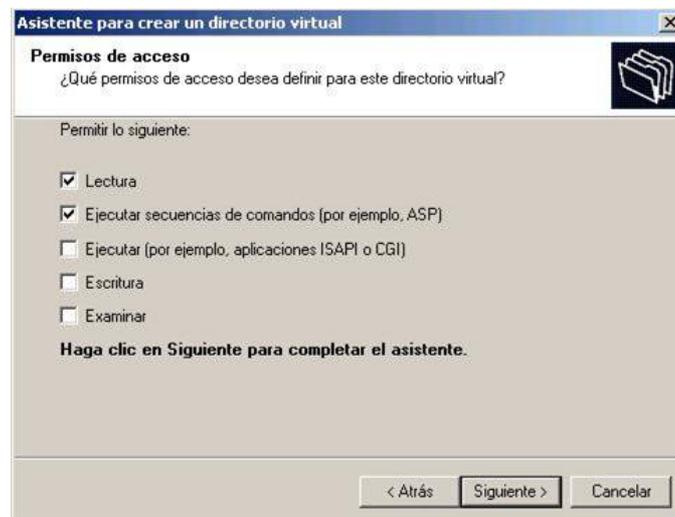


Alias del directorio virtual

Y a continuación nos pedirá la ubicación física del directorio, es decir su trayectoria y los permisos que queremos que posea.



Ubicación del directorio



Permisos del directorio virtual

Seguridad de un sitio Web

Realmente, los mayores esfuerzos de Microsoft a la hora de lanzar esta nueva versión de IIS se han centrado en la seguridad. Dejando a un lado los agujeros de seguridad que caracterizan a este software (pero sin olvidarse de ellos), IIS 5.0 incorpora nuevas y muy buenas funcionalidades referentes a la autenticación y la seguridad

Normalmente, el acceso a un servidor web (acceso a sus recursos) se lleva a cabo de forma anónima o más bien bajo la apariencia de un usuario que se crea con tal propósito en el momento de la instalación de IIS. Este usuario se denomina IUSR_nombreservidor.

Pero será conveniente limitar el acceso a ciertas zonas del servidor que contienen información privilegiada o simplemente información preparada para un usuario o máquina concreta, ya que IIS utiliza las características de seguridad de Windows 2000 y el sistema de ficheros NTFS para fijar la política de seguridad del sitio web.

Si queremos restringir el acceso a nuestro sitio web o a partes del mismo, deberemos modificar las propiedades predeterminadas en la pestaña Seguridad de directorios.



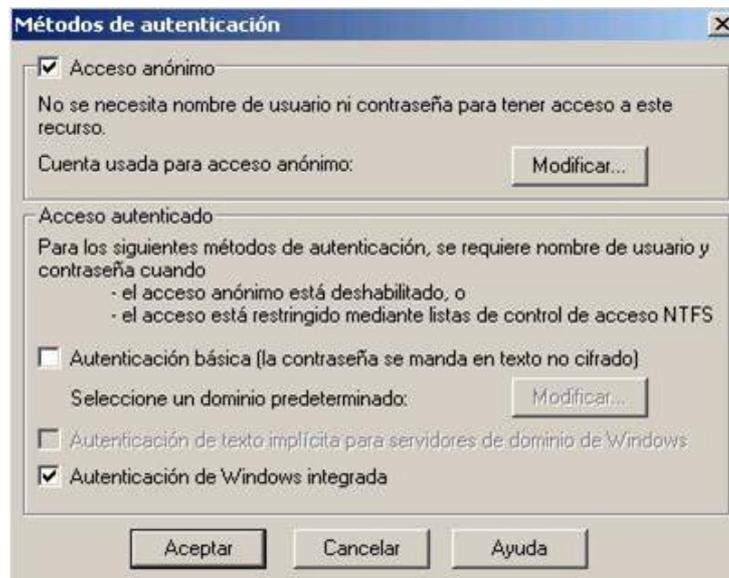
Seguridad en directorios

Si lo que deseamos es proteger una zona del servidor según el usuario que solicita el recurso, emplearemos esta propiedad, la cual nos permite definir tres métodos de autenticación

- **Autenticación básica** La autenticación básica es un método estándar, soportado por la mayoría de navegadores, que solicita al cliente un usuario y un password. El problema radica en que esa información crítica de seguridad viaja por la red sin cifrar.
- **Autenticación de texto implícita** La autenticación de texto implícita, una nueva característica de IIS 5.0, ofrece las mismas características que la autenticación básica, pero incluye una forma diferente de transmitir las credenciales de autenticación. Las credenciales de autenticación pasan por un proceso unidireccional, frecuentemente llamado hashing. El resultado de este proceso se denomina hash o código resultado del mensaje y no es factible descifrarlo. Es decir, no se puede descifrar el texto original a partir del hash.

- Autenticación de Windows integrada La autenticación de Windows integrada (anteriormente llamada NTLM o Autenticación Desafío/Respuesta de Windows NT) es un método seguro de autenticación, ya que no se envía a través de la red el nombre de usuario ni la contraseña. Al habilitar la autenticación de Windows integrada, el explorador del usuario demuestra que conoce la contraseña mediante un intercambio criptográfico con el servidor Web, en el que interviene el hashing.

En la siguiente figura se muestran como seleccionar y definir las propiedades de los diferentes métodos de autenticación.



Métodos de autenticación

Copia de seguridad y restauración de la configuración

Una de las grandes ventajas de IIS (para algunos administradores) es su configuración gráfica, debido a su interfaz intuitiva y de fácil uso. Pero no sería práctico, sino pudiéramos salvar la información de configuración para poder restaurarla después.

Siempre he pensado que lo mejor es un fichero de configuración tipo texto, donde poder ir modificando variables y así tener una copia del mismo. No hay que alarmarse debido a que IIS no ofrezca esta característica ya que si que soporta salvar la configuración en un formato propio de Windows para posteriormente restaurarla de forma sencilla.

Los pasos a realizar para hacer una copia de seguridad de la configuración actual serían los siguientes:

- En el complemento, Administración de servicios de Internet, se selecciona el icono del equipo para resaltarlo en el panel izquierdo.

- En el menú Acción, elija la opción Realizar o restaurar copia de seguridad de la configuración.
- Hacer clic en el botón Realizar copia de seguridad, que mostrará un cuadro de diálogo Realizar copia de seguridad que le permitirá definir el archivo de copia de seguridad.

De forma predeterminada, el archivo de copia de seguridad se almacena en el directorio `c:\winnt\system32\inetsrv\Metaback`

Instalación de IIS en Windows 2000

Internet Information Server (IIS) es el servidor de páginas web avanzado de la plataforma Windows. Se distribuye gratuitamente junto con las versiones de Windows basadas en NT, como pueden ser Windows 2000 Profesional o Windows 2000 Server, así como Windows XP, también en sus versiones Profesional y Server.

Estas normas de instalación son aplicables, a nivel general, a las que podemos encontrarnos en las distintas versiones de los sistemas operativos comentados antes, si bien hemos tomado Windows XP profesional para relatar los pasos y tomar las imágenes de las pantallas.

Nota: Windows 95, 98, las versiones Home, de Windows XP, y ME, de Windows 2000, no se admite la instalación de IIS. En su lugar podemos probar a instalar el Personal Web Server, que se explica en el artículo [Instalación de Personal Web Server](#).

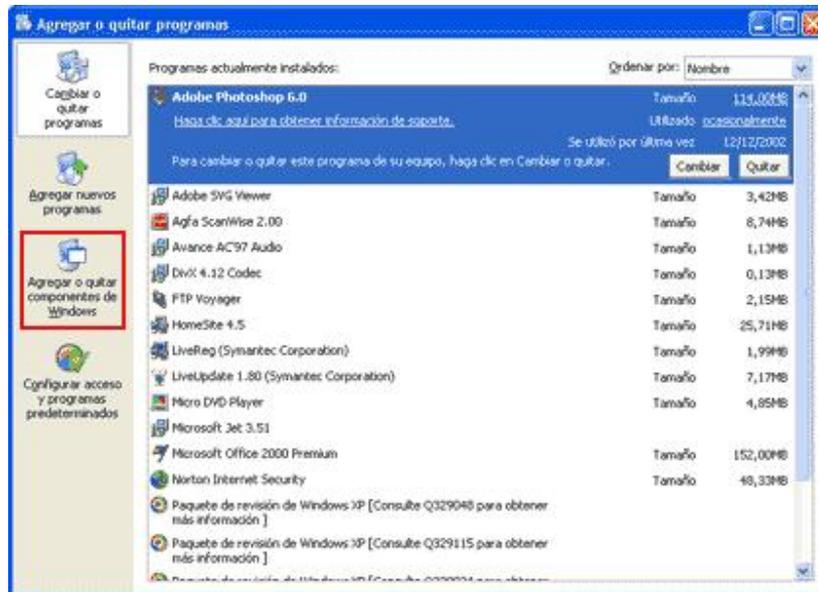
Agregar componentes adicionales de Windows

IIS se puede encontrar en el propio CD de instalación de Windows XP Profesional. Hay que acceder a la opción de "Instalar componentes opcionales de Windows" para poder cargarlo en nuestro sistema. Para ello tenemos dos opciones:

- 1) Insertar el CD de instalación de Windows y en la ventana de autoarranque que se muestra, seleccionar la opción que pone "Instalar componentes opcionales de Windows"

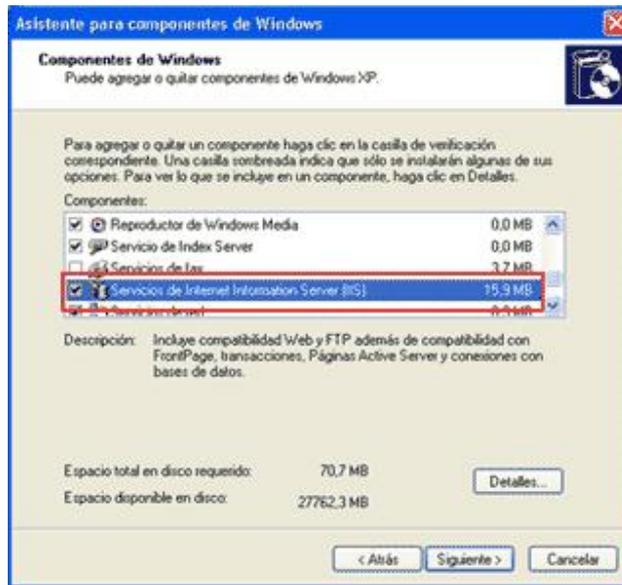


2) En el Panel de control, seleccionar la opción de "Agregar o quitar programas" y en la ventana que sale, pulsar sobre el icono de la izquierda marcado como "Seleccionar o quitar componentes de Windows".



Ahora nos muestra la ventana para seleccionar los componentes adicionales de Windows que hay disponibles. En la lista, marcamos la opción "Servicios de Internet Information Server (IIS)". Por defecto se seleccionan unos cuantos componentes, dentro de los que ofrece la instalación de IIS. Nosotros podemos elegir qué componentes deseamos instalar apretando el botón marcado como "Detalles". Entre los componentes posibles se encuentran las extensiones de Frontpage, documentación, servicios adicionales de IIS, un servidor de FTP (para la transferencia de ficheros con el servidor por FTP), incluso uno de SMTP (para el envío de correos electrónicos).

Si no sabemos qué componentes instalar podemos dejar las opciones como aparecen en un principio, pues para la mayoría de los casos serán válidas. Sólo un detalle: puede ser adecuado no instalar las extensiones de Frontpage en caso de que no pensemos que se vayan a utilizar.



Una vez hemos instalado los componentes deseados, podemos y apretar el botón de "Siguiente" para comenzar la instalación, que se alargará unos minutos.

Acceder al servidor web

Podemos acceder al servidor web para comprobar si se ha instalado correctamente IIS. Para ello simplemente debemos escribir `http://localhost` en Internet Explorer y debería aparecer una página web informando que IIS está correctamente instalado. Además, aparecerá la documentación de IIS en una ventana emergente, si es que fue instalada.



Sitio web predeterminado en IIS

Lo que se muestra cuando accedemos a `http://localhost` es el sitio web predeterminado, que se guarda en nuestro disco duro, concretamente en la carpeta `C:\inetpub\wwwroot`

Si accedemos a dicha carpeta desde Mi PC podremos ver los archivos que se están sirviendo como sitio web predeterminado. Encontraremos, entre otros archivos uno llamado "iisstart.asp" que es el que se pone en marcha al acceder a este directorio.

Colocar nuestras propias páginas

Lo lógico ahora es que deseemos colocar nuestras propias páginas web para que las sirva IIS. Si nuestro sitio web es bastante simple podríamos colocar todos los archivos dentro de la carpeta del sitio web predeterminado.

Por ejemplo, para hacer una prueba, podríamos colocar un archivo llamado "hola.asp" en la carpeta C:\inetpub\wwwroot. Para acceder a este archivo desde nuestro explorador deberíamos escribir la dirección `http://localhost/hola.asp`

Si deseamos un código simple para el archivo `hola.asp`, en el que se pruebe si está funcionando o no el motor ASP, podemos utilizar el siguiente:

```
<html>
<head>
<title>Probando ASP</title>

</head>
<body>
<%
for i=1 to 7

        response.write "<font size=" & i & ">Hola Mundo!</font><br>"

next
%>
</body>
</html>
```

Atención: No se debe acceder al archivo utilizando una ruta como esta: C:\inetpub\wwwroot\hola.asp, pues de esa manera no se estaría pasando a través del servidor Web y la página ASP no se ejecutaría.

Si tuviéramos un sitio medianamente complejo, convendría crear una carpeta dentro de `wwwroot` con el nombre de nuestro sitio y dentro podríamos colocar todos los archivos. Si el directorio tuviera una ruta como C:\inetpub\wwwroot\mi_web, accederíamos colocando esta dirección en nuestro navegador: `http://localhost/mi_web`.

Cuando se accede a ese directorio se sirve el documento por defecto que se haya configurado en el servidor web. El documento por defecto es aquel que se sirve si no se ha especificado ninguno en la ruta de acceso, es decir, si no indicamos ningún archivo en la dirección que colocamos en

Internet Explorer (una dirección como http://localhost/mi_web especifica un directorio, pero ningún archivo) se sirve el documento por defecto.

En principio, el documento por defecto en IIS está configurado a Default.asp o Default.htm. Esto quiere decir que deberíamos colocar un archivo con uno de esos nombres en nuestro directorio para que se sirva si el usuario no indica ningún nombre de archivo. Luego veremos cómo cambiar esta configuración.

Para probar todo esto, podemos crear un archivo llamado Default.asp y guardarlo en nuestro directorio C:\inetpub\wwwroot\mi_web. El código que podríamos utilizar sería por ejemplo este:

```
<html>
<head>
<title>Archivo por defecto en mi_web</title>

</head>
<body>
<h1>Archivo por defecto en mi_web</h1>

<%
for i=0 to 9

response.write i

next
%>
</body>
</html>
```

Recordar que para ver este archivo habría que componer una dirección como esta http://localhost/mi_web. Automáticamente se sirve el archivo Default.asp, aunque no se especifique nada, pues ese es el documento por defecto. También podríamos acceder al archivo especificando su ruta completa: http://localhost/mi_web/Default.asp

Nota: Podemos llamar al archivo Default.asp o bien default.asp (con mayúscula o minúscula en su inicial). Cualquiera de las dos opciones es válida, pues en Windows no se tienen en cuenta las mayúsculas y minúsculas en nombres de archivos.

Administración de IIS

Para administrar el servidor Internet Information Server en Windows XP, disponemos de un panel de control llamado "Servicios de Internet Information Server" al que podemos acceder de varias maneras.

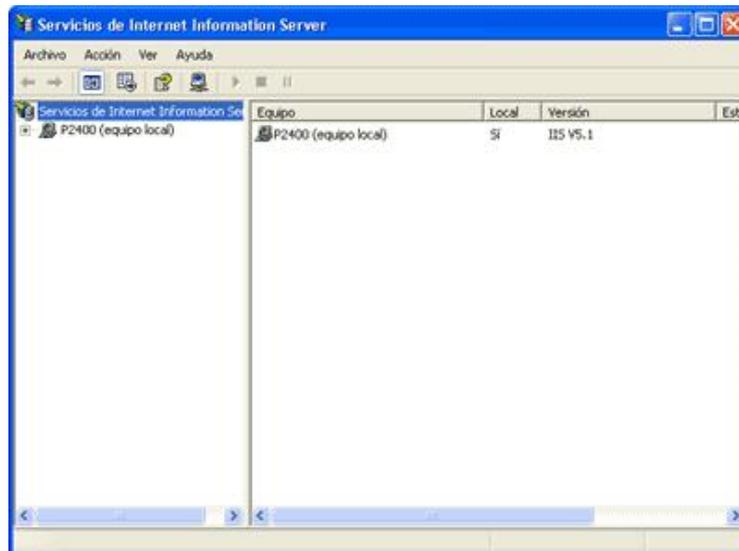
1) Pulsando con el botón derecho en MI PC y seleccionando la opción que pone "Administrar".

Esto nos abre "Microsoft Management Console" o, dicho en castellano, la "Administración de equipos". En la lista de la izquierda, en la parte de abajo aparece "Servicios y aplicaciones", entre los que encontraremos una opción buscada: "Servicios de Internet Information Server"

2) Podemos acceder desde el panel de control. Si tenemos configurada la vista clásica encontraremos un icono que pone "Herramientas administrativas" y haciendo doble clic, encontraremos el icono para administrar IIS. Si teníamos configurada la vista por categorías del panel de control (la que aparece por defecto en Windows XP) la búsqueda de la opción es un poco más compleja: Seleccionamos "Rendimiento y mantenimiento" y dentro ya encontraremos el icono de "Herramientas administrativas", al que teníamos que hacer doble clic para encontrar, entre otros, el icono para acceder a "Servicios de Internet Information Server".

3) Otra manera de acceder aparece en la ayuda de Internet Information Server. Se trata de hacer una búsqueda del archivo llamado "inetmgr.exe". Una vez localizado se puede ejecutar y aparece la consola de administración de IIS. Si se desea, se puede hacer un acceso directo a dicho archivo para no tener que buscarlo cada vez que se desee ejecutar.

Una vez hemos accedido al panel "Servicios de Internet Information Server" tenemos ante nosotros la posibilidad de configurar nuestro servidor web en muchos aspectos, por ejemplo podemos, definir el documento por defecto, crear directorios virtuales, modificar las opciones de seguridad, etc.



Definir el documento por defecto en IIS

Hablamos antes sobre el documento por defecto, que en IIS viene definido en un principio en los archivos default.asp, default.htm o index.htm. Estos archivos son muy específicos de Windows, pero no suelen utilizarse en la mayoría de los proveedores, así que es mejor que utilicemos un

documento por defecto idéntico al que utilizan la mayoría de los proveedores de hosting.

Si nuestro documento por defecto no es el mismo en local (nuestro servidor IIS) y en remoto (espacio en nuestro albergue en un servidor web conectado permanentemente a Internet), puede que tengamos que realizar algunas modificaciones en los nombres de los archivos, para que todo funcione perfectamente al subir el sitio web a Internet, lo que resultaría un engorro adicional e innecesario, de haber configurado nuestro servidor desde un principio.

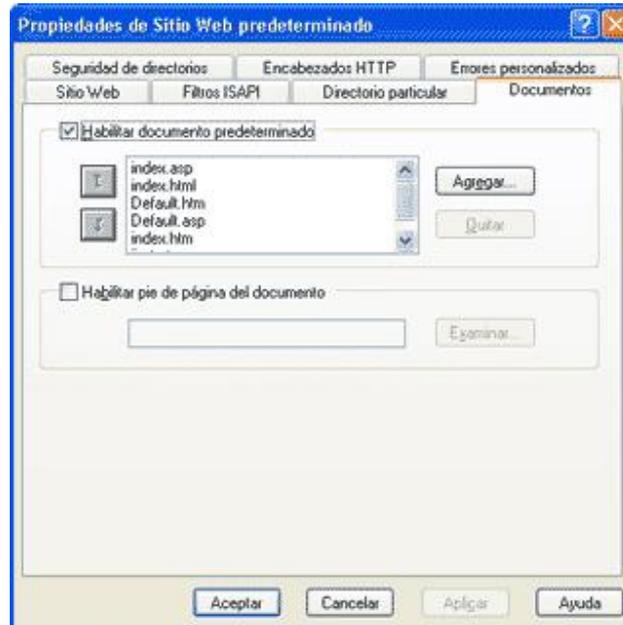
Documentos por defecto típicos son index.html, index.asp (Si es que estamos programando páginas ASP) o index.php (si es que estamos programando con PHP).

Para definir el documento por defecto debemos apretar con el botón derecho el sitio web que deseamos modificar y seleccionar la opción "Propiedades".

Entonces aparece la ventana de propiedades de este sitio, donde debemos seleccionar la pestaña marcada como "Documentos" para poder definir el documento o los documentos por defecto.



Podemos definir uno o varios documentos por defecto, de modo que, si no existe el primer archivo seleccionado como documento por defecto, se intentaría con el segundo, el tercero y cuantos haya configurados hasta que se encuentre un archivo que mostrar o se acabe la lista. Por tanto, el orden de los distintos archivos configurados como documento por defecto si importa y se puede modificar utilizando las flechas de la izquierda de la lista de posibles documentos.



Si no hay ningún archivo en el directorio cuyo nombre sea alguno de los documentos por defecto, no se mostraría ningún archivo y en su lugar recibiríamos un error 404 o el listado de ese directorio, depende de cómo esté configurado IIS para este caso.

Directorios virtuales en IIS

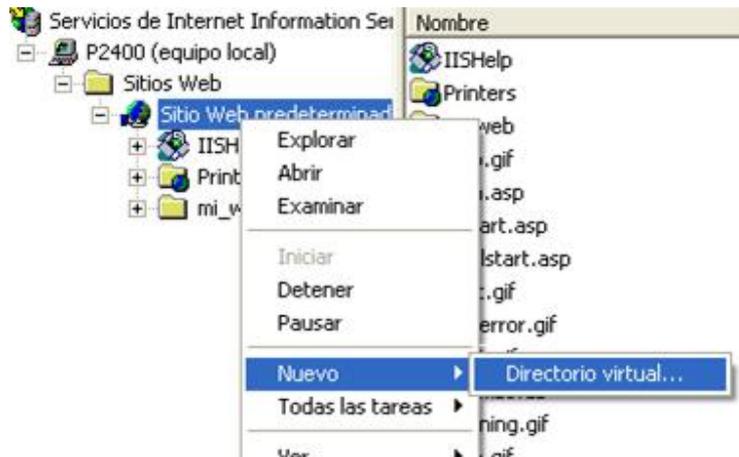
Un directorio virtual es un directorio del servidor que no está dentro del directorio de publicación habitual, es decir, un directorio que no depende de `C:\inetpub\wwwroot` pero que sí que se puede acceder a través del servidor web como si estuviera dentro de dicho directorio.

Como ya habíamos comentado, para acceder a nuestro IIS necesitamos escribir una dirección como esta: `http://localhost`. Así se accede al directorio `C:\inetpub\wwwroot`, que es llamado directorio particular. El directorio virtual se accede con algo como `http://localhost/directorio_virtual`, pero no tiene porque existir una correspondencia en disco de este directorio dentro de la carpeta de publicación, es decir, no tiene porque existir el directorio `C:\inetpub\wwwroot\directorio_virtual`, sino que dicho directorio podría estar en cualquier otro sitio de nuestro disco duro, por ejemplo `C:\mis_paginas`.

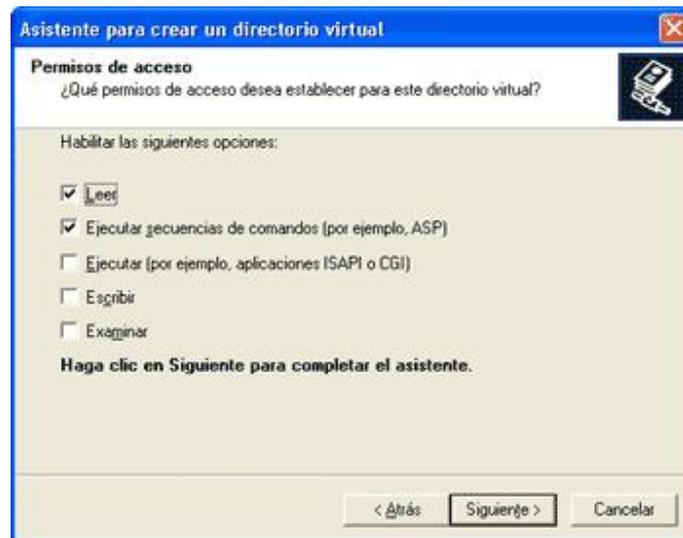
Los directorios virtuales se pueden mapear hacia otro directorio de nuestros discos duros o incluso a otro directorio situado en otro ordenador de la red.

Crear un directorio virtual

Para definir un directorio virtual se puede pulsar con el botón derecho del ratón sobre el sitio web en el que queremos definirlo y seleccionar la opción "Nuevo > Directorio Virtual...". Entonces aparece un asistente que nos guiará paso a paso en el proceso.



El primer paso del asistente nos pregunta el "alias" o nombre lógico que queremos darle al directorio. El segundo paso nos pide la localización física de ese directorio en nuestro disco duro o en la red local. Finalmente nos solicita los permisos que deseamos asignar a ese directorio. El permiso de lectura y el de ejecución de secuencias de comandos (Por ejemplo, ASP) suelen ser suficientes para la mayoría de los casos.



Una vez finalizado el asistente queda creado el directorio virtual y podremos accederlo a través del alias que hayamos seleccionado.

Apéndice C

Sección 1

```
CREATE DATABASE sistema_ie
ON
( NAME = sistema_ie,
  FILENAME = 'C:\sistema_ie\DATA\sistema_ie.mdf' )
LOG ON
( NAME = 'sistema_ie_log',
  FILENAME = 'C:\sistema_ie\DATA\sistemaie_log.ldf' )

GO
RESTORE DATABASE Sistema_ie FROM DISK = 'C:\sistema_ie\instala.sar'
EXEC sp_addlogin 'Sistema_ie','SAR_ojos','Sistema_ie'
USE Sistema_ie
EXEC sp_changedbowner 'Sistema_ie'
GO
```

Sección 2

```
cambio();
cmb_dep.Clear;
statusbar1.Panels[1].Text:= 'Usuarios Conectados';
tb_dependencia.Active:=True;
for i:=0 to tb_dependencia.RecordCount-1 do
begin
  cmb_dep.Items.Add(tb_dependencia.Fields[0].Text);
  tb_dependencia.Next;
end;
tb_dependencia.Active:=False;

cmb_so.Clear;
tb_sistema.Active:=True;
for i:=0 to tb_sistema.RecordCount-1 do
begin
  cmb_so.Items.Add(tb_sistema.Fields[0].Text);
  tb_sistema.Next;
end;
tb_sistema.Active:=False;

cmb_ubica.Clear;
tb_ubica.Active:=True;
```

```
for i:=0 to tb_ubica.RecordCount-1 do
begin
    cmb_ubica.Items.Add(tb_ubica.Fields[0].Text);
    tb_ubica.Next;
end;
tb_ubica.Active:=False;
```

```
cmb_nic.Clear;
tb_nic.Active:=True;
for i:=0 to tb_nic.RecordCount-1 do
begin
    cmb_nic.Items.Add(tb_nic.Fields[0].Text);
    tb_nic.Next;
end;
tb_nic.Active:=False;
```

```
cmb_te.Clear;
tb_te.Active:=True;
for i:=0 to tb_te.RecordCount-1 do
begin
    cmb_te.Items.Add(tb_te.Fields[0].Text);
    tb_te.Next;
end;
tb_te.Active:=False;
```

```
edit2.OnChange(Sender);
tb_conectado.Active:=True;
p_conecta.Visible:=True;
```

Sección 3:

```
var strc:string;
begin
    strc:=edit2.Text+'*';
    case radiogroup2.ItemIndex of
    0: tb_conectado.Filter:='dirIP = '+QuotedStr(strc);
    1: tb_conectado.Filter:='inventario = '+QuotedStr(strc);
    2: tb_conectado.Filter:='MACaddress = '+QuotedStr(strc);
    3: tb_conectado.Filter:='usuario = '+QuotedStr(strc);
    4: tb_conectado.Filter:='dependencia = '+QuotedStr(strc);
    end;
    if edit2.Text="" then
        tb_conectado.Filter:="";
```

Sección 4

```
tb_dependencia.SetKey;
case radiogroup1.ItemIndex of
  0:begin
    tb_dependencia.IndexName:='IX_'+radiogroup1.Items.Strings[0];
    tb_dependencia.FindNearest([txtbus_dep.text]);
  end;
  1:begin
    tb_dependencia.IndexName:='IX_rango'+radiogroup1.Items.Strings[1];
    tb_dependencia.FindNearest([txtbus_dep.text]);
  end;
  2:begin
    tb_dependencia.IndexName:='IX_'+radiogroup1.Items.Strings[2];
    tb_dependencia.FindNearest([txtbus_dep.text]);
  end;
end;
```

Sección 5

```
cambio();
statusbar1.Panels[1].Text:='Catalogo Dependencias';
txtbus_dep.OnChange(sender);
cmbubica.Clear;
cmbdep.Clear;
tb_ubica.Active:=True;
for i1:=0 to tb_ubica.RecordCount-1 do
begin
  cmbubica.Items.Add(tb_ubica.Fields[0].Text);
  tb_ubica.Next;
end;
tb_ubica.Active:=False;

tb_dependencia.Active:=True;
for i1:=0 to tb_dependencia.RecordCount-1 do
begin
  cmbdep.Items.Add(tb_dependencia.Fields[0].Text);
  tb_dependencia.Next;
end;
p_dependencia.Visible:=True;
```

.

Sección 6

```
var stra:string;
begin
  stra:=edit1.Text+'*';
case radiogroup3.ItemIndex of
  0: tb_nt.Filter:='Usuario = '+QuotedStr(stra);
  1: tb_nt.Filter:='nuser = '+QuotedStr(stra);
end;
if edit1.Text="" then
tb_nt.Filter:="";
```

Sección 7

```
cambio();
statusbar1.Panels[1].Text:='Usuarios Windows NT - Dominio GEH';
edit1.OnChange(sender);
P_NT.Visible:=true;
tb_nt.Active:=true;
```

Sección 8

```
var stra:string;
begin
  stra:=edit3.Text+'*';
case radiogroup4.ItemIndex of
  0: tb_mail.Filter:='Nombre = '+QuotedStr(stra);
  1: tb_mail.Filter:='Cuenta = '+QuotedStr(stra);
  2: tb_mail.Filter:='Cargo = '+QuotedStr(stra);
  3: tb_mail.Filter:='Dependencia = '+QuotedStr(stra);
  4: tb_mail.Filter:='IP = '+QuotedStr(stra);
end;
if edit3.Text="" then
  tb_mail.Filter:="";
```

Sección 9

```
cambio();
edit3.OnChange(Sender);
statusbar1.Panels[1].Text:='Usuarios de Correo Electronico';
cmb_depm.Clear;
tb_dependencia.Active:=True;
for i:=0 to tb_dependencia.RecordCount-1 do
```

```
begin
    cmb_depdm.Items.Add(tb_dependencia.Fields[0].Text);
    tb_dependencia.Next;
end;
tb_dependencia.Active:=False;

cmb_usuariom.Clear;
tb_user.Active:=True;
for i:=0 to tb_user.RecordCount-1 do
begin
    cmb_usuariom.Items.Add(tb_user.Fields[0].Text);
    tb_user.Next;
end;
tb_user.Active:=False;

tb_mail.Active:=True;
p_correo.visible:=True;
```

Sección 10

```
cambio();
cmb_deps.Clear;
statusbar1.Panels[1].Text:='Servicios';
tb_dependencia.Active:=True;
for i222:=0 to tb_dependencia.RecordCount-1 do
begin
    cmb_deps.Items.Add(tb_dependencia.Fields[0].Text);
    tb_dependencia.Next;
end;
tb_dependencia.Active:=False;

cmb_tserv.Clear;
tb_servicio.Active:=True;
for i222:=0 to tb_servicio.RecordCount-1 do
begin
    cmb_tserv.Items.Add(tb_servicio.Fields[0].Text);
    tb_servicio.Next;
end;
tb_servicio.Active:=False;

tb_sers.Active:=True;
datetimepicker1.Date:=strtodate(Editf_sol.Text);
datetimepicker2.Date:=strtodate(Editf_pet.Text);
p_serv.Visible:=True;
```

Sección 11

```
tb_sers.FindNearest([edit7.text]);
```

Sección 12

```
if OpenFileDialog1.Execute then  
  dbmemo4.Lines.LoadFromFile(OpenDialog1.FileName);
```

Sección 13

```
cambio();  
statusbar1.Panels[1].Text:='Preguntas Frecuentes';  
tb_tip.Active:=true;  
p_tips.Visible:=True;
```

Sección 14

```
cambio();  
memo1.Clear;  
statusbar1.Panels[1].Text:='Reportes';  
if FileExists('reporte.txt') then  
  begin  
    memo1.Lines.LoadFromFile('reporte.txt');  
    edit4.Text:=memo1.Lines.Strings[0];  
    edit5.Text:=memo1.Lines.Strings[1];  
    edit6.Text:=memo1.Lines.Strings[2];  
  end  
else  
  begin  
    edit4.Text:="";  
    edit5.Text:="";  
    edit6.Text:="";  
    MessageDlg('Haga click en el boton MODIFICAR para anexar los titulos que se incluiran en los  
reportes.',mtinformation, [mbok], 0);  
  end;  
p_reporte.visible:=true;  
treeview1.SetFocus;
```

Sección 15

```
SpeedButton9.Enabled:=true;  
SpeedButton8.Enabled:=False;  
edit4.SetFocus;
```

Sección 16

```
var FF:textFile;
begin
  AssignFile(FF, 'reporte.txt');
  Rewrite(FF);
  Writeln(FF, edit4.text);
  Writeln(FF, edit5.text);
  Writeln(FF, edit6.text);
  CloseFile(FF);

SpeedButton8.Enabled:=true;
SpeedButton9.Enabled:=False;
```

Sección 17

```
var i232:integer;
begin
  cambio1();
  case TreeView1.Selected.AbsoluteIndex of
    0,1:begin
      lblcmb.Visible:=false;
      combobox1.Visible:=False;
      end;
    2:begin
      form2.SpeedButton7.Top:=120;
      form2.SpeedButton10.Top:=120;
      combobox1.Visible:=true;
      lblcmb.Caption:=treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text;
      lblcmb.Visible:=True;
      combobox1.Clear;
      if NOT(tb_dependencia.Active) then
        tb_dependencia.Active:=true;
        for i232:=0 to tb_dependencia.RecordCount-1 do
          begin
            combobox1.Items.Add(tb_dependencia.Fields[0].Text);
            tb_dependencia.Next;
          end;
        tb_dependencia.Active:=False;
        combobox1.Visible:=True;
        combobox1.Text:=combobox1.Items.Strings[0];
        form3.tb_conectado.Filter := 'dependencia = '+QuotedStr(combobox1.Text);
      end;
    3:begin
      form2.SpeedButton7.Top:=120;
```

```
form2.SpeedButton10.Top:=120;
combobox1.Visible:=true;
lblcmb.Caption:=treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text;
lblcmb.Visible:=True;
combobox1.Clear;
if NOT(tb_te.Active) then
  tb_te.Active:=true;
  for i232:=0 to tb_te.RecordCount-1 do
    begin
      combobox1.Items.Add(tb_te.Fields[0].Text);
      tb_te.Next;
    end;
  tb_te.Active:=False;
  combobox1.Visible:=True;
  combobox1.Text:=combobox1.Items.Strings[0];
  form3.tb_conectado.Filter := 'tipo_equipo = '+QuotedStr(combobox1.Text);
end;
4:begin
  lblcmb.Visible:=false;
  combobox1.Visible:=false;
end;
7:begin
form2.SpeedButton7.Top:=120;
form2.SpeedButton10.Top:=120;
combobox1.Visible:=true;
lblcmb.Caption:=treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text;
lblcmb.Visible:=True;
combobox1.Clear;
if NOT(tb_user.Active) then
  tb_user.Active:=true;
  for i232:=0 to tb_user.RecordCount-1 do
    begin
      combobox1.Items.Add(tb_user.Fields[0].Text);
      tb_user.Next;
    end;
  tb_user.Active:=False;
  combobox1.Visible:=True;
  combobox1.Text:=combobox1.Items.Strings[0];
  form3.tb_mail.Filter := 'usuario = '+QuotedStr(combobox1.Text);
end;
8:begin
form2.SpeedButton7.Top:=120;
form2.SpeedButton10.Top:=120;
  lblcmb.Caption:=treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text;
  combobox1.Visible:=true;
```

```
lblcmb.Visible:=True;
combobox1.Clear;
if NOT(tb_dependencia.Active) then
  tb_dependencia.Active:=true;
  for i232:=0 to tb_dependencia.RecordCount-1 do
    begin
      combobox1.Items.Add(tb_dependencia.Fields[0].Text);
      tb_dependencia.Next;
    end;
  tb_dependencia.Active:=False;
  combobox1.Visible:=True;
  combobox1.Text:=combobox1.Items.Strings[0];
  form3.tb_mail.Filter := 'Dependencia = '+QuotedStr(combobox1.Text);
end;
10:begin
form2.SpeedButton7.Top:=88;
form2.SpeedButton10.Top:=88;
  combobox1.Visible:=true;
  lblcmb.Caption:=treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text;
  lblcmb.Visible:=True;
  combobox1.Clear;
  if NOT(tb_user.Active) then
    tb_user.Active:=true;
    for i232:=0 to tb_user.RecordCount-1 do
      begin
        combobox1.Items.Add(tb_user.Fields[0].Text);
        tb_user.Next;
      end;
    tb_user.Active:=False;
    combobox1.Visible:=True;
    combobox1.Text:=combobox1.Items.Strings[0];
    form3.tb_mail.Filter := 'usuario = '+QuotedStr(combobox1.Text)+ 'AND Internet = '+chr(39)+'SI'+chr(39);
  end;

12:begin
form2.SpeedButton7.Top:=120;
form2.SpeedButton10.Top:=120;
  lblcmb.Caption:=treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text;
  combobox1.Visible:=true;
  lblcmb.Visible:=True;
  combobox1.Clear;
  if NOT(tb_servicio.Active) then
    tb_servicio.Active:=true;
    for i232:=0 to tb_servicio.RecordCount-1 do
```

```
begin
  combobox1.Items.Add(tb_servicio.Fields[0].Text);
  tb_servicio.Next;
end;
tb_servicio.Active:=False;
combobox1.Visible:=True;
combobox1.Text:=combobox1.Items.Strings[0];
form3.tb_sers.Filter := 'servicio = '+QuotedStr(combobox1.Text);
end;
end;
```

Sección 18

```
if TreeView1.Selected.AbsoluteIndex <> -1 then
begin
  titulos();
  case TreeView1.Selected.AbsoluteIndex of
  0:begin
    form3.tb_dependencia.active:=true;
    form3.rep_dep.Preview;
    form3.tb_dependencia.active:=False;
  end;
  1:begin
    form3.tb_conectado.active:=true;
    form3.rep_conectados.Preview;
    form3.tb_conectado.active:=False;
  end;
  2:begin
    form3.QRDBText3.DataField:='tipo_equipo';
    form3.QRLabel10.Caption:=' Tipo de Equipo';
    form3.tb_conectado.active:=true;

    form3.QRLabel7.Caption:=form3.QRLabel7.Caption+treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text+' : '+combobox1.Text;
    form3.rep_conectados.Preview;
    form3.tb_conectado.active:=False;
    form3.tb_conectado.Filter :='';
    form3.QRLabel7.Caption:='Equipos conectados por ';
  end;

  3:begin
    form3.QRDBText3.DataField:='dependencia';
    form3.QRLabel10.Caption:=' Dependencia';
```

```
form3.tb_conectado.active:=true;
form3.QRLabel7.Caption:=form3.QRLabel7.Caption+'Tipo : '+combobox1.Text;
form3.rep_conectados.Preview;
form3.tb_conectado.active:=False;
form3.QRLabel7.Caption:='Equipos conectados por ';
form3.tb_conectado.Filter :="";
end;
4:begin
form3.tb_nt.active:=true;
form3.rep_userNT.Preview;
form3.tb_nt.active:=False;
end;
7:begin
form3.QRDBText15.DataField:='Usuario';
form3.QRLabel27.Caption:='Tipo de Usuario';
form3.tb_mail.active:=true;

form3.query1.Active:=true;

form3.QRLabel19.Caption:=form3.QRLabel19.Caption+' por
'+treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text+' : '+combobox1.Text;
form3.rep_int.Preview;
form3.tb_mail.active:=False;
form3.QRLabel19.Caption:='Usuarios';
form3.tb_mail.Filter :="";
form3.query1.Active:=False;
end;
8:begin
form3.QRDBText15.DataField:='Usuario';
form3.QRLabel27.Caption:='Tipo de Usuario';
form3.tb_mail.active:=true;
form3.QRLabel19.Caption:=form3.QRLabel19.Caption+' por
'+treeview1.Items.Item[TreeView1.Selected.AbsoluteIndex].Text+' : '+combobox1.Text;
form3.rep_mail.Preview;
form3.tb_mail.active:=False;
form3.QRLabel19.Caption:='Usuarios';
form3.tb_mail.Filter :="";
end;
12:begin
form3.tb_sers.active:=true;
form3.QRLabel28.Caption:=form3.QRLabel28.Caption+' : '+combobox1.Text;
form3.rep_ser.Preview;
form3.tb_sers.active:=False;
form3.QRLabel28.Caption:='Servicio';
form3.tb_sers.Filter :="";
```

```
end;  
end  
end  
else  
    MessageDlg('Seleccione el reporte que desea ver',mtinformation, [mbok], 0);
```

Sección 19

```
case TreeView1.Selected.AbsoluteIndex of  
0:begin  
    ARWordReport1.DataSetList.Items[0].Dataset:=tb_dependencia;  
    tb_dependencia.Active:=true;  
    ARWordReport1.Filename:='frm_dep.doc';  
    ARWordReport1.DataSetList.Items[0].Dataset:=tb_dependencia;  
    ARWordReport1.DataSetList.Items[0].DSName:='dependencia';  
    ARWordReport1.CustTagValues.Values['date1']:=edit4.Text;  
    ARWordReport1.CustTagValues.Values['date2']:=edit5.Text;  
    ARWordReport1.CustTagValues.Values['date3']:=datetostr(now());  
    ARWordReport1.CustTagValues.Values['date4']:= 'Dependencias';  
    ARWordReport1.Preview;  
    tb_dependencia.Active:=false;  
end;  
1:begin  
    ARWordReport1.DataSetList.Items[0].Dataset:=tb_conectado;  
    tb_conectado.Active:=true;  
    ARWordReport1.Filename:='frm_conecta.doc';  
    ARWordReport1.DataSetList.Items[0].DSName:='conectado';  
    ARWordReport1.CustTagValues.Values['date1']:=edit4.Text;  
    ARWordReport1.CustTagValues.Values['date2']:=edit5.Text;  
    ARWordReport1.CustTagValues.Values['date3']:=datetostr(now());  
    ARWordReport1.CustTagValues.Values['date4']:= 'Usuarios Conectados';  
    ARWordReport1.Preview;  
    tb_conectado.Active:=false;  
end;  
2:begin  
    ARWordReport1.DataSetList.Items[0].Dataset:=tb_conectado;  
    tb_conectado.Active:=true;  
    ARWordReport1.Filename:='frm_con_dep.doc';  
    ARWordReport1.DataSetList.Items[0].DSName:='conectado';  
    ARWordReport1.CustTagValues.Values['date1']:=edit4.Text;  
    ARWordReport1.CustTagValues.Values['date2']:=edit5.Text;  
    ARWordReport1.CustTagValues.Values['date3']:=datetostr(now());
```

```
ARWordReport1.CustTagValues.Values['date4']:='Usuarios Conectados: '+combobox1.Text;
ARWordReport1.Preview;
tb_conectado.Active:=false;
end;
3:begin
ARWordReport1.DataSetList.Items[0].Dataset:=tb_conectado;
tb_conectado.Active:=true;
ARWordReport1.FileName:='frm_con_te.doc';
ARWordReport1.DataSetList.Items[0].DSName:='conectado';
ARWordReport1.CustTagValues.Values['date1']:='edit4.Text';
ARWordReport1.CustTagValues.Values['date2']:='edit5.Text';
ARWordReport1.CustTagValues.Values['date3']:='datetostr(now());
ARWordReport1.CustTagValues.Values['date4']:='Usuarios conectados por Tipo de Equipo:
'+combobox1.Text;
ARWordReport1.Preview;
tb_conectado.Active:=false;
end;
4:begin
ARWordReport1.DataSetList.Items[0].Dataset:=tb_nt;
tb_nt.Active:=true;
ARWordReport1.FileName:='frm_nt.doc';
ARWordReport1.DataSetList.Items[0].DSName:='unt';
ARWordReport1.CustTagValues.Values['date1']:='edit4.Text';
ARWordReport1.CustTagValues.Values['date2']:='edit5.Text';
ARWordReport1.CustTagValues.Values['date3']:='datetostr(now());
ARWordReport1.CustTagValues.Values['date4']:='Usuarios de Windows NT';
ARWordReport1.Preview;
tb_nt.Active:=false;
end;

6:begin
ARWordReport1.DataSetList.Items[0].Dataset:=tb_mail;
tb_mail.Active:=true;
ARWordReport1.FileName:='frm_mail_tu.doc';
ARWordReport1.DataSetList.Items[0].DSName:='mail';
ARWordReport1.CustTagValues.Values['date1']:='edit4.Text';
ARWordReport1.CustTagValues.Values['date2']:='edit5.Text';
ARWordReport1.CustTagValues.Values['date3']:='datetostr(now());
ARWordReport1.CustTagValues.Values['date4']:='Usuarios de Correo: '+combobox1.Text;
ARWordReport1.Preview;
tb_mail.Active:=false;
end;

7:begin
ARWordReport1.DataSetList.Items[0].Dataset:=tb_mail;
```

```
tb_mail.Filter := 'Internet = '+QuotedStr('Si');
tb_mail.Active:=true;
ARWordReport1.FileName:='frm_mail_www.doc';
ARWordReport1.DataSetList.Items[0].DSName:='mail';
ARWordReport1.CustTagValues.Values['date1']:=edit4.Text;
ARWordReport1.CustTagValues.Values['date2']:=edit5.Text;
ARWordReport1.CustTagValues.Values['date3']:=datetostr(now());
ARWordReport1.CustTagValues.Values['date4']:= 'Usuarios de Correo';
ARWordReport1.Preview;
tb_mail.Active:=false;
end;

8:begin
ARWordReport1.DataSetList.Items[0].Dataset:=tb_mail;
tb_mail.Active:=true;
ARWordReport1.FileName:='Frm_mail_dep.doc';
ARWordReport1.DataSetList.Items[0].DSName:='mail';
ARWordReport1.CustTagValues.Values['date1']:=edit4.Text;
ARWordReport1.CustTagValues.Values['date2']:=edit5.Text;
ARWordReport1.CustTagValues.Values['date3']:=datetostr(now());
ARWordReport1.CustTagValues.Values['date4']:= 'Usuarios de Correo: '+combobox1.Text;
ARWordReport1.Preview;
tb_mail.Active:=false;
end;

9:begin
ARWordReport1.DataSetList.Items[0].Dataset:=tb_sers;
tb_sers.Active:=true;
ARWordReport1.FileName:='Frm_serv.doc';
ARWordReport1.DataSetList.Items[0].DSName:='serv';
ARWordReport1.CustTagValues.Values['date1']:=edit4.Text;
ARWordReport1.CustTagValues.Values['date2']:=edit5.Text;
ARWordReport1.CustTagValues.Values['date3']:=datetostr(now());
ARWordReport1.CustTagValues.Values['date4']:= 'Servicio: '+combobox1.Text;
ARWordReport1.Preview;
tb_sers.Active:=false;
end;
end;
```

Seccion 20

```
cambio();
statusbar1.Panels[1].Text:='Catalogos';
Catalogos2.Visible:=true;
form2.tb_sistema.Active:=not False;
```

```
form2.tb_servicio.Active:=not False;
form2.tb_te.Active:=not False;
form2.tb_user.Active:=not False;
```

```
sp_reg_cat.Visible:=True;
sp_avz_cat.Visible:=True;
```

```
p_catalogos.Visible:=true;
```

Sección 21

```
procedure cambio;
```

```
begin
```

```
if form2.tb_mail.Active then
  form2.tb_mail.Active:=false;
form2.tb_mail.Filter:='';
```

```
if form2.tb_sistema.Active then
  form2.tb_sistema.Active:=false;
form2.tb_sistema.Filter:='';
```

```
if form2.tb_ubica.Active then
  form2.tb_ubica.Active:=false;
form2.tb_ubica.Filter:='';
```

```
if form2.tb_nic.Active then
  form2.tb_nic.Active:=false;
form2.tb_nic.Filter:='';
```

```
if form2.tb_servicio.Active then
  form2.tb_servicio.Active:=false;
form2.tb_servicio.Filter:='';
```

```
if form2.tb_dependencia.Active then
  form2.tb_dependencia.Active:=false;
form2.tb_dependencia.Filter:='';
```

```
if form2.tb_conectado.Active then
  form2.tb_conectado.Active:=false;
form2.tb_conectado.Filter:='';
```

```
if form2.tb_user.Active then
  form2.tb_user.Active:=false;
form2.tb_user.Filter:='';
```

```
if form2.tb_te.Active then
  form2.tb_te.Active:=false;
form2.tb_te.Filter:="";

if form2.tb_nt.Active then
  form2.tb_nt.Active:=false;
form2.tb_nt.Filter:="";

if form2.tb_sers.Active then
  form2.tb_sers.Active:=false;
form2.tb_sers.Filter:="";

if form2.tb_tip.Active then
  form2.tb_tip.Active:=false;

form2.Catalogos2.Visible:=false;
form2.p_serv.visible:=False;
form2.p_correo.visible:=False;
form2.P_NT.Visible:=false;
form2.p_catalogos.Visible:=false;
form2.p_conecta.Visible:=false;
form2.p_dependencia.Visible:=false;
form2.p_reporte.visible:=False;
form2.p_tips.Visible:=False;
form2.sp_reg_cat.Visible:=False;
form2.sp_avz_cat.Visible:=False;

form2.ComboBox1.Visible:=False;
form2.lblcmb.visible:=false;
form2.lblcmb.Visible:=false;

end;
```

Sección 22

```
if MessageDlg('¿Desea realmente Instalar la Base de Datos nueva?',mtConfirmation, [mbYes,
mbNo], 0) = mrYes then
  begin
    instala.Execute;
    MessageDlg('La instalacion de la base de datos se ha realizado
satisfactoriamente.',mtInformation, [mbOk], 0);
  end
else
  MessageDlg('Ha cancelado la instalacion de la Base de Datos',mtConfirmation, [mbOk], 0);
```

CODIGO DEL PROCEDIMIENTO DE INSTALACIÓN:

```

CREATE DATABASE sistema_ie
ON
( NAME = sistema_ie,
  FILENAME = 'C:\sistema_ie\DATA\sistema_ie.mdf' )
LOG ON
( NAME = sistema_ie_log,

  FILENAME = 'C:\sistema_ie\DATA\sistemaie_log.ldf' )
GO
RESTORE DATABASE Sistema_ie FROM DISK = 'C:\sistema_ie\instala.sar'
EXEC sp_addlogin 'Sistema_ie','SAR_ojos','Sistema_ie'
USE Sistema_ie
EXEC sp_changedbowner 'Sistema_ie'
GO

```

Sección 23

```

if label30.Caption<>" then
begin
if NOT(FileListBox1.Visible) then
begin
  bachup.CommandText:='BACKUP DATABASE SISTEMA_IE TO DISK =
'+chr(39)+label30.Caption+chr(39);
  bachup.Execute;
  MessageDlg('Se ha completado la Copia',mtInformation, [mbOk], 0)
end
else
begin
  bachup.CommandText:='RESTORE DATABASE SISTEMA_IE FROM DISK =
'+chr(39)+label30.Caption+chr(39);
  bachup.Execute;
  MessageDlg('Se ha completado la restauracion',mtInformation, [mbOk], 0)
end;
end
else
  MessageDlg('Asegurese que este bien escrito el nombre del archivo o que la ruta sea
correcta',
            mtInformation,[mbOk], 0);

```

Sección 24

```
procedure TForm1.GeneralHttpEngine1ExecRequest(Sender: TObject);
var i,fea,fea1:integer;
    SS : TStringList;
    ojos_feos:string;
begin
if (FormVar('REC',">") AND (fl=0)then
begin
    id:=FormVar('USERID',"");
    pas:=FormVar('PASS',"");

    TrimRight(pas);
    table1.Active:=True;
    table1.First;

    for i:=0 to table1.RecordCount-1 do
        begin
            if (id=UID.Text) AND (pas=pass.Text) then
                begin
                    putline('<center>');
                    Putline('<hr>');
                    Putline('<font size=4>'+depende.Text+'</font><hr>');
                    HttpmemoFilter2.Put;
                    putline('</center>');
                    fl:=10;
                    exit;
                end;
            table1.Next;
        end;

    if i>(table1.RecordCount-1) then
        begin
            Putline('<center><hr>');
            Putline('<font size=4>¡¡¡ERROR!!! Nombre de Usuario o Contraseña no
validos...</font></center>');
            end;
            table1.Active:=false;
        end
    else
        if (FormVar('REC1',">") AND (fl2=0) then
            begin
                table1.Active:=True;
                table1.First;
```

```

id:=FormVar('USERID','');
pas:=FormVar('PASS','');
for i:=0 to table1.RecordCount-1 do
if (id=UID.Text) AND (pas=pass.Text) then
begin
Putline('<hr>');
ojos:=FormVar('INPUT4','');
case strtoint(ojos) of
1:begin
tb_conecta.Active:=True;
query1.SQL.Clear;

query1.SQL.Add('SELECT usuario Usuario, dirIP Direccion_IP, inventario
Inventario, equipo Equipo, tipoSO Sistema_Operativo, TipoNIC Tarjeta_de_Red, nodo
Nodo, tipo_equipo Tipo_de_Equipo, MACaddress MAC_Address FROM conectado
WHERE Dependencia = '+chr(39)+depende.Text+chr(39));
Query1.Active:=true;
HttpDBGrid1.Put;
tb_conecta.Active:=False;
Query1.Active:=false;
end;
2:begin
tb_mail.Active:=True;
query2.SQL.Clear;
query2.SQL.Add('SELECT Nombre Nombre_de_Usuario, usuario
Tipo_de_Usuario, Cuenta Cuenta, pass Password, Cargo Cargo, IP Direccion_IP FROM
Mail WHERE Dependencia = '+chr(39)+depende.Text+chr(39));
Query2.Active:=true;
HttpDBGrid2.Put;
Query2.Active:=false;
tb_mail.Active:=False;
end;
3: begin
tb_mail.Active:=True;
query2.SQL.Clear;
query2.SQL.Add('SELECT Nombre Nombre_de_Usuario, usuario
Tipo_de_Usuario, Cargo Cargo, IP Direccion_IP FROM Mail WHERE Internet =
'+chr(39)+'Si'+chr(39)+' AND Dependencia = '+chr(39)+depende.Text+chr(39));
Query2.Active:=true;
HttpDBGrid2.Put;
Query2.Active:=false;
tb_mail.Active:=False;
end;

```



```

    else
    begin
        HttpMemoFilter1.Put;
    end;
end;

if EntryName='FORMAS' then
begin
    tb_tip.Active:=True;
    tb_tip.First;
    for i1:=0 to tb_tip.RecordCount-1 do
begin
    putline('<OPTION>'+dedit1.text+'</OPTION>');
    tb_tip.Next; end;
    tb_tip.Active:=False;
end
end

```

Sección 25: Códigos HTML usados

CODIGO HttpmemoFilter1

```

<html>
<center>
%HEADER%
<form method="post" action"%script_name%">
User ID : </td><td><input name="USERID" value="%USERID%"><br>
Password : </td><td><input type="PASSWORD" name="PASS" value="%PASS%"><br><br>
<td align=center><input type=submit NAME="REC" value="ENVIAR..."></td><br>
</center>
</center>
</form>
</body>
</html>
<BR><P>
<BR><P>
<BR><P>
<BR><P>
<BR><P>
<font size=2>
<BR><P> <center>
Para cualquier comentario favor de comunicarse con alguna de las siguientes personas:<P>
<B><I>
Ing. Lissette Hernández <BR>
Ing. Horacio Rubio <BR>
M. A. Nelly Cerón<BR>

```

```

</I></B>
<P>a la ext. 6789 o enviar mensaje al correo: <A HREF="mailto: admin_redes@edo-
hidalgo.gob.mx">admin_redes@edo-hidalgo.gob.mx</A>
</center></font>

```

CODIGO HttpmemoFilter2

```

<form method="post" action="%script_name%">
  Equipos Conectados<INPUT NAME="INPUT4" SIZE=140 TYPE=RADIO
VALUE="1"><BR><P>
  Usuarios de Correo<INPUT NAME="INPUT4" SIZE=140 TYPE=RADIO VALUE="2"><BR><P>
  Usuarios de Internet<INPUT NAME="INPUT4" SIZE=140 TYPE=RADIO VALUE="3"><BR><P>
  Solución a preguntas frecuentes:<INPUT NAME="INPUT4" SIZE=140 TYPE=RADIO
VALUE="4"><BR>

```

```

<input type=hidden name="USERID" value="%USERID%">
<input type=hidden name="INPUT5" value="%INPUT5%">
<input type=hidden name="PASS" value="%PASS%">
  <TD>%INPUT4%</TD>
<BR><BR>
</TR><td align=center><input type=submit NAME="REC1" value="VER..."></td><br>
<BR><BR>
</center>
</form>

```

CODIGO HttpmemoFilter3

```

<form method="post" action="%script_name%">
<center>
  <TD>
    <SELECT NAME="INPUT5" SIZE=5 MULTIPLE>
      %FORMAS%
    </SELECT></TD>
  <TD>%INPUT5%</TD>

<BR><BR>
</TR><td align=center><input type=submit NAME="REC2" value="VER..."></td><br>
</center>
</form>
<BR><P>
<BR><P>
<BR><P> <center>
Si desea hacer alguna sugerencia o aportación enviar mensaje al correo: <A HREF="mailto:
admin_redes@edo-hidalgo.gob.mx">admin_redes@edo-hidalgo.gob.mx</A>
</center>

```

Apéndice D

Introducción a la Ingeniería de Software

Es la ciencia que ayuda a elaborar sistemas con el fin de que sea económico, fiable y funcione eficientemente sobre las maquinas reales.

La ingeniería de software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de 3 elementos clave: métodos, herramientas y procedimientos, estos facilitan al gestor a controlar el proceso de desarrollo de software y suministra a los que practique dicha ingeniería las bases para construir software de alta calidad.

Los métodos de la ingeniería de software. Suministran el cómo construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requerimientos del sistema y del software, diseño de procedimientos algorítmicos, codificación, prueba y mantenimiento.

Los métodos de la ingeniería de software introducen frecuentemente una notación especial orientada al lenguaje o gráfica y a un conjunto de criterios para la calidad del software.

Las herramientas de ingeniería de software. Suministran un soporte automático o semiautomático para los métodos. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software llamado ingeniería de software asistido por computadora, por mencionar alguna de estas herramientas existen las herramientas CASE.

CASE[(Ingeniería de software asistida por computadora) computer aided software engineering]. Combina del software, hardware y base de datos para crear un entorno de la ingeniería de software.

Las herramientas son como voy a aplicar los métodos para tener un desarrollo.

Las herramientas de ingeniería de software son los métodos necesarios para desarrollar el sistema.

Los procedimientos de la ingeniería de software. Son la cola que paga a los métodos y herramientas, facilita un desarrollo racional y oportuno del software de computadoras.

Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas que se requieren y los controles que ayuden asegurar, la calidad y coordinar los cambios y las guías que facilitan a los gestores de software establecer el desarrollo.

PARADIGMAS

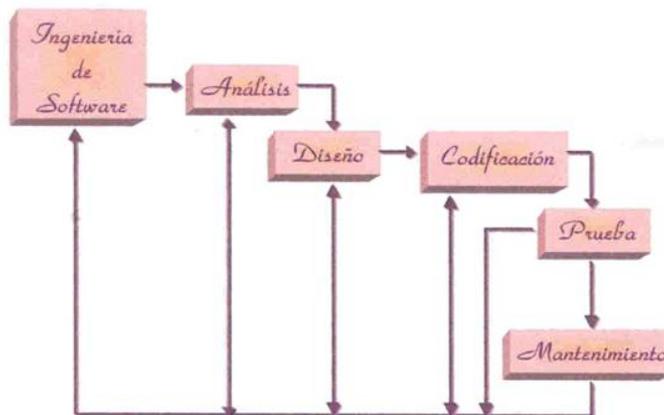
La ingeniería de software esta compuesta por una serie de pasos de abarcan los metodos, las herramientas y los procedimientos antes mencionados. estos pasos se denominan frecuentemente paradigmas de la ingenieria de software. La elección de un paradigma para la ingeniería de software se lleva a cabo de acuerdo con la naturaleza del proyecto y de la aplicación, los metodos, herramientas a usar, los controles y entregas requeridos.

Gama de paradigmas de la ingeniería de software:



PARADIGMA CICLO DE VIDA DEL SOFTWARE

Este fue el modelo inicial planteado para organizar el proceso de desarrollo, aunque antiguo tiene vigencia en algunos proyectos o como parte de otros modelos, da la medida de los pasos tradicionales de cualquier modelo: análisis, diseño, codificación, prueba y mantenimiento.



Ingeniería y análisis del sistema. Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo todos los requerimientos o elementos del sistema y luego asignando algún subconjunto de estos requerimientos al software; esta versión del sistema es esencial cuando el software debe interrelacionarse con otros elementos tales como hardware, personas y bases de datos.

La ingeniería y análisis del sistema abarcan los requerimientos globales a un nivel de sistema con una pequeña cantidad de análisis y diseño a nivel superior. Además de un análisis costo beneficio del sistema es decir si toda la inversión que se hará para el sistema conviene a los beneficios que traerá el mismo.

Análisis de los requerimientos del software. El proceso de recoger los requerimientos se centra y se intensifica especialmente en esta etapa. Para comprender la naturaleza de los programas que hay que construir. El ingeniero de software debe comprender el dominio de la información del software, así como la función, rendimiento e interfaces requeridas. En esta etapa los requerimientos del sistema se documentan y se analizan con el cliente.

Diseño. El diseño del software es realmente un proceso multipaso que se enfoca sobre 3 atributos del programa.

- estructura de datos
- arquitectura de software
- detalle procedimental

El diseño traduce los requerimientos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes que comience la codificación. Como los requerimientos y el diseño que se documentan forman parte de la configuración del software.

Codificación. El diseño debe traducirse en una forma legible. El paso de la codificación ejecuta la tarea de establecer la etapa de diseño legible para la maquina, si el diseño se ejecuta de una manera detallada la codificación puede realizarse mecánicamente.

Prueba. Una vez que se ha generado el código, comienza la prueba del programa, la prueba se enfoca sobre la lógica interna del software asegurando que todas las sentencias se han probado y sobre las funciones externas estoy realizando pruebas para asegurar que la entrada definida producirá los resultados que realmente se requieren.

Mantenimiento. El software sufrirá indudablemente cambios después que se le entregue al cliente los cambios ocurrirán debido a que se han encontrado errores, causados por cambios del entorno externo por ejemplo un cambio solicitado debido al cambio del Sistema Operativo o dispositivos periféricos, o debido que el cliente requiere aumentos en las funciones del sistema. El mantenimiento del software se aplica cada uno de los pasos precedentes del ciclo de vida a un programa existente en lugar de uno nuevo.

El ciclo de vida clásico es el mas viejo y más ampliamente usado paradigma en la ingeniería de software. Sin embargo con el paso de unos cuantos años se han producido criticas al paradigma, incluso por seguidores activos que cuestionan su aplicabilidad a todas las situaciones. Entre los problemas que se presentan algunas veces cuando se aplica este paradigma se encuentran:

Los proyectos reales raramente siguen el flujo secuencial que propone el modelo, siempre ocurre iteraciones y se crea problemas en la aplicación del paradigma.

Normalmente es difícil para el cliente establecer explícitamente el principio todos los requerimientos del sistema. El ciclo de vida clásico requiere esto y tiene dificultades para acomodar posibles incertidumbres que puedan existir al comienzo de dichos proyectos.

El cliente debe tener paciencia. Una versión funcional del programa no esta disponible hasta las etapas finales del desarrollo del proyecto. Un error importante no detectado hasta que el programa este en funcionamiento puede ser desastroso.

Los responsables del desarrollo se retrasan innecesariamente.

Las fases genéricas que caracterizan el proceso de software (definición, desarrollo y mantenimiento) son aplicables a todo software. El problema es seleccionar el modelo de proceso apropiado para la ingeniería de software que debe aplicarse en el proyecto.

El gestor del proyecto debe decidir que método de proceso es el mas adecuado para el proyecto, después debe definir un plan preliminar basado en un conjunto de actividades estructurarles validas para cualquier proyecto. Una vez establecido el plan preliminar, empieza la descomposición del proceso. Es decir, se debe crear un plan completo reflejando las tareas requeridas a las personas para cubrir las actividades estructurales.

Maduración Del Problema Y El Proceso.

La planificación del proyecto empieza con la maduración del problema y del proceso. Todas las funciones que se deben tratar dentro de un proceso de ingeniería por el equipo de software deben pasar por el conjunto de actividades estructurales que se han definido por una organización de software. Asuma que la organización ha adoptado el siguiente conjunto de actividades estructurales:

- *Comunicación con el cliente:* tareas requeridas para establecer una comunicación eficiente entre el desarrollo y el cliente.
- *Planificación:* Tareas requeridas para definir los recursos, la planificación temporal del proyecto y cualquier información relativa a el.
- *Análisis de riesgo:* Tareas requeridas para valorar los riesgos técnicos y de gestión.
- *Ingeniería:* Tareas requeridas para construir una o mas representaciones de la aplicación.

- *Construcción y entrega:* Tareas requeridas para construir, probar, instalar y proporcionar asistencia al usuario. (ejemplo: documentación y entrenamiento).
- *Evaluación del cliente:* Tareas requeridas para obtener la información de la opinión del cliente basadas en la evaluación de las representaciones de software creadas durante la fase de ingeniería e implementadas durante la fase de ingeniería e implementadas durante la fase de instalación.

Los miembros del equipo que trabajan en cada función aplicaran todas las actividades estructurales. Cada función principal del problema se lista en una columna de la izquierda. Las actividades estructurales se listan en la fila de arriba. las tareas de trabajo de la ingeniería de software se introducirán en la fila siguiente. El trabajo de gestor del proyecto es estimar los requisitos de recursos para cada celda de la matriz, poner fechas de inicio y finalización para las tareas asociadas con cada celda y los productos a fabricas como consecuencia de cada celda.

Conclusiones

El S.A.R. es un sistema que nos permite una administración de usuarios de una manera mas automatizada y por ende una agilización en los procesos de la información para tenerla siempre disponible tanto para los usuarios como para el sitio Web.

Con todo esto se aprendieron muchas cosas, desde la instalación de un sitio Web hasta el manejo del DBMS como lo es SQL Server. Con ello podemos observar que podemos combinar varias tecnologías que están al alcance de nuestras manos para poder realizar software con excelente manufactura y eficiencia.

En esta versión del S.A.R. existen sus ventajas y desventajas, que podemos dejar de observar ya que como toda aplicación de software tiene sus limitantes.

Algunas ventajas:

- Utilización de Software y Hardware que ya se tiene.
- Sencillez al momento de realizar captura de datos
- Agilización de procesos para la generación de reportes.
- Consulta remota de datos mediante el sitio Web.

Como hemos mencionado existen limitantes entre las cuales podemos observar las siguientes:

- Tiempo de vida de Software y Hardware
- Seguridad mayor en la sección Web, es decir, agregar certificados y encriptación de datos.

Con todo ello podemos decir que ha sido una gran experiencia el poder realizar dicho Software para una institución tan grande e importante como lo es el Gobierno del Estado, esto me permitió tener contacto con tecnologías que hasta cierto punto eran un mito y que se pueden utilizar para agilizar procesos y tener toda la información de manera segura y confiable.

GLOSARIO

A

ADO: ActiveX Data Objects

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.

ASP: Active Server Pages

B

BD: Base de datos

C

CGI: Common Gateway Interface

D

DBMS: Manejador de Bases de Datos

DBA: Administrador de base de datos

DML: Data Manipulation Language (Lenguaje de Manipulación de Datos)

DCL: Data Control Language (Lenguaje de Control de Datos)

DDL: Data Definition Language (Lenguaje de Definición de Datos)

I

INFORMIX: Informix-4GL ofrece herramientas para crear menús, formularios de entrada de datos y generadores de listados. Será necesario definir estas mismas herramientas manteniendo, a ser posible, la sintaxis original. En principio se generarán aplicaciones que funcionen en modo texto, dejando para una futura ampliación la generación de aplicaciones en entornos gráficos.

IMAP: Internet Message Access Protocol: Su finalidad es la misma que la de POP, pero el funcionamiento es diferente

J

JDBC: Java DataBase Connectivity

O

ODBC: Open DataBase Connectivity

ORACLE: Oracle es un sistema de administración de base de datos (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:
Soporte de transacciones.
Estabilidad.
Escalabilidad.
Es multiplataforma.

P

POP: Post Office Protocol: Se utiliza para obtener los mensajes guardados en el servidor y pasárselos al usuario

PHP: Hipertext Preprocesor

S

SQL: Lenguaje de Consulta Estructurado

SSL: (Secure Sockets Layer) es un protocolo diseñado por la empresa Netscape Communications, que permite cifrar la conexión, incluso garantiza la autenticación. Se basa en la criptografía asimétrica y en el concepto de los certificados.

Servidor ftp: una computadora que sirve cualquier tipo de archivos, a través del Protocolo de Transferencia de Ficheros a clientes FTP o a navegadores web que lo soporten.

SMTP: Simple Mail Transport Protocol: Es el protocolo que se utiliza para que dos servidores de correo intercambien mensajes

X

XML: Extensible Markup Language (lenguaje de marcado extensible)

BIBLIOGRAFÍA

- [Delphi, 1993] Delphi 3.0.
Editorial Prentice may
México, D.F, 1993
- [Encarta, 1999] Encarta® 99, Enciclopedia Microsoft®. ©
Microsoft Corporation.
1993-1998
- [Análisis y Diseño de Sistemas, 1997] Kendall & Kendall
Análisis y Diseño de Sistemas.
Editorial Prentice Hall (Pearson Educación)
México, D.F, 1997
- [Análisis y Diseño de Sistemas, 2000] James A. Senn
Análisis y Diseño de Sistemas
Segunda Edición, Mc Graw Hill
Abril 2000.
- [SQL Server 2000, 2001] Delaney, Kalen
A fondo, Microsoft SQL Server 2000
Editorial Mc Graw Hill
Aravaca, Madrid, 2001
- [Ingeniería del Software,2000] Roger S. Pressman
Ingeniería del Software 4ª Edición
Mc Graw Hill
- [Procesamiento de Bases de Datos, 1996] David M. Kroenke
Procesamiento de Bases de Datos
Quinta edición, Prentice Hall
México 1996
- [Sistemas de Bases de Datos, 1990] Alice Y. H. Tsai
Sistemas de Bases de Datos, Administración y
uso
Editorial Prentice Hall
México 1990

- [Fundamentos de Bases de Datos, 1995] Henry F. Korth, Abraham Silberschatz
Fundamentos de Bases de Datos
Segunda edición
Editorial Mc.Graw Hill
- [Servidores de Bases de Datos para Internet con CGI, 1996] Rowe, Jeff
Servidores de Bases de Datos para Internet con CGI,
New Riders Publishing
1996.
- [Ingeniería de software un enfoque práctico, 4a. Edición.1997] Roger S. Pressman
Ingeniería de software un enfoque práctico, 4a. Edición.
Ed. Mc Graw Hill
1997.