# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

## FACULTAD DE INGENIERÍA

"Desarrollo de una solución basada en Web para el PROGRAMA DE MANEJO AMBIENTAL DE LOS RESIDUOS PROVENIENTES DE DESAZOLVE DEL SISTEMA DE DRENAJE DEL DF Y DE LAS PRESAS, ASÍ COMO DE PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES MUNICIPALES "

Propuesta de TESIS

QUE PARA OBTENER EL GRADO DE

Ingeniero en Computación

PRESENTAN:

*Osnaya Medrano Pedro*

*Cruz Sánchez Sayabil Yac*

*Velázquez Alvarez Mauricio*

*AGRADECIMIENTOS*

*A la Universidad Nacional Autónoma de México, por darnos la oportunidad de pertenecer a su comunidad, por permitirnos desarrollarnos en sus instalaciones y por la formación que recibimos en sus aulas.*

*A la Facultad de Ingeniería, por permitirnos ser uno más de sus alumnos, por entregarnos, historia, maestros, instalaciones y recursos, formándonos como profesionistas, como ingenieros.*

*Al Instituto de Ingeniería, por brindarnos la oportunidad de integrarnos en sus actividades, desarrollando nuestras habilidades y conocimientos, por mostrarnos una nueva faceta de la ingeniería.*

# ÍNDICE

## ÍNDICE DE FIGURAS

# CAPÍTULO 1 FUNDAMENTOS TEÓRICOS

Aún cuando nuestro proyecto tiene como finalidad la construcción e implementación de una solución basada en Web, es necesario presentar un marco teórico adecuado para señalar las ventajas y desventajas del uso del modelo basado en Web. Es necesario plantear los antecedentes de los diversos modelos de datos, teniendo en cuenta que no se deberá de confundir con modelado de los objetos de la base de datos o características técnicas del manejador o servidor de base de datos seleccionado para nuestro proyecto. Debido a que las tecnologías que hoy existen nos permiten un mayor número de posibilidades para establecer una solución basada en Web, los fundamentos teóricos necesarios, un estudio adecuado de las necesidades de nuestro cliente, sus recursos y los aspectos funcionales de las herramientas de desarrollo que podamos utilizar, son los elementos básicos para proponer una solución integral; tanto las nuevas tecnologías como las tecnologías robustas ya probadas por varios años en la industria, nos proporcionan modelos diferentes y provistos de sus propias variantes con respecto a los modelos teóricos. Una nueva herramienta requiere de un modelado con aspectos particulares para poder ser funcional de acuerdo a sus innovaciones; estos temas serán tratados con detalle en capítulos posteriores.

La revisión de los fundamentos teóricos termina, con las características de los diversos paradigmas de programación existentes y una descripción más detallada del paradigma de programación seleccionado para este proyecto.

## 1.1 MODELO DE APLICACIÓN

La construcción de una solución requiere planear correctamente y no solo dejarse llevar por las tendencias del mercado o por las opiniones de grandes instituciones. Los tres principales modelos existentes permiten dar soluciones confiables a cada problema. Dentro de una organización, los sistemas de información se apoyan en una infraestructura lógica, esta infraestructura ha estado ligada en el pasado al propio modelo de la organización. Tradicionalmente las organizaciones han tenido una estructura centralizada y jerárquica, estructurada en departamentos con cometidos concretos, las relaciones entre los distintos departamentos dentro de la jerarquía se encuentran perfectamente definidas.

El modelo actual de organización, por el contrario, se articula en unidades más operativas y autónomas, que funcionan por cumplimiento de objetivos, existen menos niveles jerárquicos y las relaciones que existen entre las distintas unidades son más directas y pueden variar con el tiempo. Sin embargo, por otra parte se tiende a centralizar los datos corporativos que son importantes desde el punto de vista estratégico. Debido a esta evolución, la infraestructura informática se ha dividido históricamente en dos tipos de modelos en extremos opuestos y un tercero que mezcla ambos conceptos:

- Modelo centralizado en la que existe un servidor central, donde residen todos los datos y procesamiento de los mismos.
- Modelo distribuido, donde el procesamiento se encuentra distribuido en diferentes máquinas y los datos pueden estar centralizados en diferentes servidores.
- Modelo basado en Web, Cuenta con uno o varios servidores, los cuales pueden almacenar, procesar y distribuir, la información entre los diversos clientes y entre los servidores.

Sus ventajas y desventajas se ven resumidos en la Tabla 1-1, aun cuando existen muchas variantes tecnológicas que parecen generar mezclas entre los diversos modelos, es conveniente afirmar que estos modelos representan un universo muy amplio de propuestas. Un ejemplo, lo representa el modelo basado en Web, el cual es para muchos autores una actualización tecnológica del modelo centralizado. Al ser un requisito de este proyecto la construcción de una solución basada en Web, es necesario conocer los tres modelos para garantizar que hacemos un uso correcto de recursos materiales y tecnológicos.

### 1.1.1 ENTORNO CENTRALIZADO

Nace en un entorno tradicional de organización, con estructura centralizada y jerárquica, dividida en departamentos. Cada departamento tiene actividades muy concretas, las relaciones que pueda establecer con otros departamentos se encuentran definidas y limitadas y suelen realizarse a través de la jerarquía empresarial. El sistema informático es único y estaba relacionado principalmente con el departamento administrativo financiero para la realización de nóminas, compras, entre otros, la funcionalidad se encuentra concentrada en un servidor central al que sólo tienen acceso los usuarios del departamento correspondiente.

Un ambiente centralizado muestra la necesidad de una estrecha comunicación entre los servidores y los clientes, permitiendo el desempeño de las tareas de los clientes con los programas y/o datos que se encuentran en los servidores. A continuación se describen las características generales que se requieren para un eficiente desempeño en un ambiente de servicios y recursos centralizados -Ortal Robert-.

## 1.2 CARACTERÍSTICAS DE UN AMBIENTE CENTRALIZADO

- **Servicio:** El ambiente Centralizado presenta una estrecha relación, los procesos se ejecutan en equipos de cómputo distintos al del cliente donde el servidor proporciona uno o más servicios, mientras que los clientes son consumidores de los servicios.
- **Recursos compartidos:** Un servidor atiende a varios clientes en forma ordenada permitiendo el acceso a los recursos que se encuentran compartidos, los cuales pueden ser servicios, archivos, sistemas de almacenamiento, u otro sistema de cómputo que proporcione un servicio.
- **Protocolos:** Bajo una relación de muchos a uno, el servidor establece la comunicación con varios clientes, los cuales inician las peticiones de servicios o recursos mediante un grupo de reglas específicas para poder ser atendidos por el servidor, el cual atiende a cada uno de los solicitantes.
- **Unificación:** La idea del ambiente centralizado es la de ejecutar las tareas sin importar la plataforma o el hardware en donde se lleva a cabo la consulta o acceso a recursos o servicios que brinda el servidor.
- **Transparencia de localización**: El servidor es un proceso que se ejecuta o reside en uno o más equipos de cómputo dentro de un ambiente de red, el cual es localizado por la redirección que se lleva a cabo para poder ingresar a sus recursos y servicios.
- **Escalabilidad:** La escalabilidad marca en el sistema centralizado un crecimiento en el entorno tanto de parte del servidor como del cliente, debido a los avances tecnológicos que se presenten y a las necesidades que se demandan por parte de los usuarios. Siendo el crecimiento tanto vertical como horizontal.
- **Integridad:** Se encuentra enfocada a los servicios que presta el servidor y a los datos del servidor, en cómo se encuentran relacionados de forma centralizada para poder responder de forma lógica y consistente a las peticiones de los clientes que solicitan aplicaciones o datos -Robert Orta-

### 1.2.1 CARACTERÍSTICAS FÍSICAS

- Único servidor corporativo dimensionado para soportar todos los procesos de la organización, todos los datos y las posibles comunicaciones con otras instancias de replicación.
- Una base de datos robusta, donde residen todos los datos, impresoras y terminales.

### 1.2.2 CARACTERÍSTICAS LÓGICAS

- Ejecución de todos los procesos en el servidor central.
- Si la empresa cuenta con una dispersión geográfica y dispone de una red de comunicaciones, todos los puestos de trabajo están conectados al servidor central formando una "estrella".

### 1.2.3 PRINCIPALES VENTAJAS

- Alto rendimiento en transacciones.
- Alta disponibilidad.
- Entorno probado y personal experimentado.
- Control total del servidor, al ser éste único y residente en el centro de proceso de datos.
- Concentración de todo el personal de explotación y administración del sistema en un único centro de proceso de datos.

### 1.2.4 PRINCIPALES INCONVENIENTES

Alto costo de los servidores, al requerir mucho poder de procesamiento, para dar servicio a todos los usuarios que estén conectados y gran espacio en disco para albergar todos los datos.

Alta dependencia de las comunicaciones. En caso de caída de una línea, todos los puestos de trabajo dependientes de dicha línea quedan inoperantes.

Interfaces de usuario de caracteres (no gráficos) y, por lo tanto, poco amigables (Situación que en los últimos año ha cambiado).

Arquitecturas propietarias (Situación que en los últimos años ha cambiado).

## 1.3 ARQUITECTURAS PARTICULARES

### 1.3.1 SERVIDOR DE ARCHIVOS

Un servidor de Archivos proporciona la cualidad de distribuir a los clientes el uso de documentos, archivos o carpetas bajo el esquema de resolver las peticiones que los clientes demandan. El esquema de trabajo de un Servidor de Archivos hace referencia a que este mismo es el que atiende a las llamadas y peticiones por medio de un servicio activo el cual es consultado por los clientes los cuales poseen las aplicaciones que modifican o requieren de los archivos alojados en los servidores. Como se muestra en la Figura 1-1

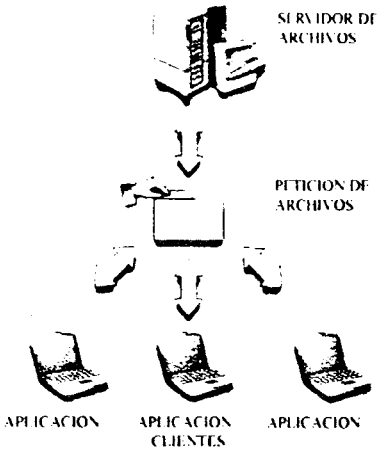



**Figura 1-1 Vista típica de un servidor de archivos**

Donde una computadora de escritorio, es típicamente el cliente que accede a los servidores, siendo este un grupo de equipos de cómputo más robustos (controlados por un servidor central) en cuanto a sus componentes de hardware y en muchas ocasiones de software también.

### 1.3.2 SERVIDORES DE BASE DE DATOS

El esquema de un Servidor de Base de Datos de la Figura 1-2 muestra la interacción entre el cliente y el servidor, se puede observar que el cliente es quien hace una consulta de los datos que se encuentran alojados en el servidor, este a su vez muestra como resultado final la búsqueda de los datos solicitados por el cliente. Este proceso se lleva a cabo de forma más óptima y de mejor desempeño bajo el uso de las aplicaciones correspondientes entre los datos solicitados y el cliente que posee las aplicaciones para su procesamiento. Este proceso es más eficiente debido a que se encuentran distribuidos todos los sistemas que intervienen para el procesamiento de una gran cantidad de datos.

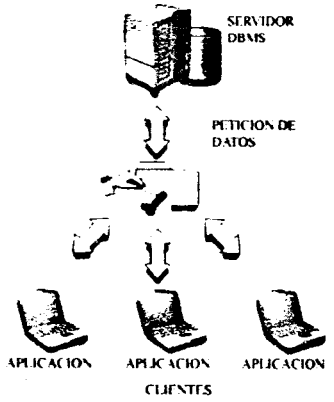



**Figura 1-2 Vista típica de un servidor de base de datos.**

### 1.3.3 SERVIDOR DE TRANSACCIONES

El proceso de transacción entre los servidores que contienen los datos y los clientes que realizan el procesamiento remoto de los datos, se lleva a cabo por medio de llamadas de los clientes hacia el servidor el cual agrupa las tareas de conexión y administración manteniendo a los usuarios en una tarea de constante comunicación del estado de los datos, cambios, ingreso de nuevos elementos, borrado, lectura, consulta, importar, entre otras tareas. El esquema de trabajo se ejemplifica en la Figura 1-3.

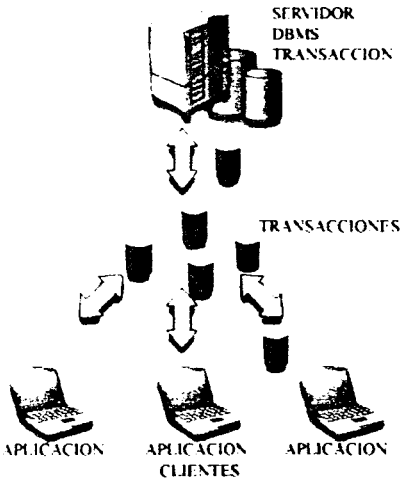



**Figura 1-3 Vista típica de un servidor de transacciones.**

### 1.3.4 SERVIDOR DE APLICACIONES

Un servidor de aplicaciones mantiene la relación centralizada con una nueva forma en cuanto al desempeño de los clientes ya que ellos mantienen las transacciones entre los servidores, pero estos últimos son los que desempeñan las tareas que los clientes requieren, siendo estas aplicaciones las que se ejecutan dentro del servidor con sus recursos como tal y deja al cliente aplicaciones que mantienen la comunicación remota para poder interactuar con las aplicaciones del servidor. No solo las bases de datos son las que muestran este desempeño de petición de actividades y resultados, ya que las nuevas tecnologías tienden a un trabajo de datos en servidores y/o clientes bajo el empleo de una red de comunicaciones.

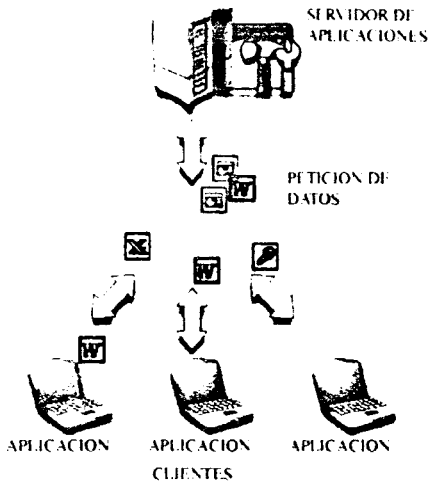



**Figura 1-4 Vista típica de un servidor de aplicaciones.**

En la Figura 1-4 se muestra como las aplicaciones son distribuidas a los clientes por medio de la petición de datos.

## 1.4 MODELO DISTRIBUIDO

Con los nuevos modelos organizativos, en los que la empresa se divide en unidades más o menos autónomas que establecen relaciones más definidas y directas entre sí, aparecen entornos informáticos departamentales adecuados a las necesidades de cada departamento en concreto. Interconecta los lugares que tienen recursos computacionales, para capturar y almacenar datos, procesarlos y enviar datos e información a otros sistemas. El rango de recursos varía, algunos lugares utilizan terminales, otros cuentan con computadoras personales, otros incluso, grandes sistemas de cómputo. En este modelo no existe el requisito de que todo el equipo sea del mismo fabricante, de hecho se espera que estén implicadas varias marcas o modelos de hardware, permitiendo al usuario tener el equipo más adecuado a sus necesidades.

Todos los equipos de cómputo (reciben el nombre de nodos en el procesamiento distribuido) tienen la capacidad de capturar y procesar datos en donde ocurran los eventos. En otras palabras, si un lugar específico usa computadoras personales, los usuarios capturan y procesan datos en ellas, reciben respuestas rápidas a sus consultas, almacenan datos en el sistema y preparan reportes cuando se necesitan. Sin embargo, también pueden transmitir datos o reportes desde su sistema a otro enlazado en la red, compuesta por todos los sistemas interconectados.

### 1.4.1 COMPONENTES DE UN SISTEMA DISTRIBUIDO

- Múltiples componentes de procesamiento de propósito general.

- Pueden asignarse tareas específicas a los sistemas de procesamiento sobre una base dinámica.
- Los sistemas no necesitan ser de una misma marca o características.
- Los nodos de procesamiento individual tienen su propio sistema operativo, el cual está diseñado para la computadora específica.
- Distribución física de los componentes. Las computadoras y otras unidades de procesamiento están separadas físicamente. Interactúan entre sí por medio de una red de comunicaciones.
- Transparencia del sistema. Los usuarios no conocen la ubicación de un componente en el sistema distribuido o dato alguno de su fabricante, modelo, sistema operativo local, velocidad o tamaño.
- El sistema operativo de los servidores lleva a cabo todas las actividades que implican la ubicación física y atributos de procesamiento para satisfacer la demanda del usuario.
- Papel dual de los componentes. Los componentes individuales de procesamiento pueden operar independientemente del marco de trabajo del sistema distribuido.

### 1.4.2 NO SON CLASIFICACIÓN COMO SISTEMAS DISTRIBUIDOS

- Una computadora multifuncional grande, que distribuye el procesamiento entre varios procesadores de entrada/salida y periféricos.
- Un procesador primario, que controla las comunicaciones del sistema al cual fue añadido.
- Un conjunto de terminales remotas, que recogen y transmiten datos a un sistema anfitrión.
- La interconexión de varias computadoras anfitrionas, que transmiten mensajes y llevan a cabo funciones y tareas exclusivas.
- La diferencia de una red de computadoras y un sistema distribuido es que en una red de computadoras el usuario se conecta explícitamente con otra máquina, ejecutando tareas remotas.

#### 1.4.2.1 CARACTERÍSTICAS DE LA DISTRIBUCIÓN

- Multiproceso (concurrencia): El hardware permite el progreso simultáneo de varias actividades (varios procesadores, con memoria local, entre otros componentes.).
- Interconexión: Permite la comunicación entre las actividades.
- Relación: Uso compartido de recursos, información, entre otros.
- Fallo independiente: Permite buscar soluciones resistentes en caso de fallo.

#### 1.4.2.2 PROPIEDADES

- Nombre único: un nombre es válido en todo el sistema.
- Acceso global: los mismos métodos actúan en objetos, en cualquier parte del sistema.
- Seguridad global: autenticación y acceso uniformes en todo el sistema.
- Disponibilidad global: funcionamiento correcto en presencia de fallos parciales.
- Gestión global: posibilidad de gestión centralizada del sistema.

### 1.4.3 CARACTERÍSTICAS FUNCIONALES

- Cada usuario trabaja con su equipo local independiente, con lo que se obtiene mejores tiempos de respuesta.
- Los recursos necesarios que no estén disponibles sobre un equipo local o una estación de trabajo dedicada, pueden ser obtenidos del servidor indicado a través de la red de comunicaciones.

#### 1.4.3.1 CARACTERÍSTICAS FÍSICAS

- Los servidores pueden no estar concentrados en una misma área, mantienen conexión por medio de una red de comunicaciones.
- Cada equipo de cómputo sobre la red tiene la capacidad de servir a las necesidades de los usuarios locales de forma independiente a los demás equipos.
- También proporciona acceso a otros elementos de la red o a servidores centrales.
- Toma importancia la red de comunicación de datos.

#### 1.4.3.2 CARACTERÍSTICAS LÓGICAS

- Las tareas más complejas o de carácter estratégico para la organización se mantienen en los servidores centrales.
- Las tareas de complejidades medias o específicas para un determinado grupo de usuarios, se distribuyen entre las máquinas locales de ese grupo.
- Los equipos de cómputo se pueden ajustar a las necesidades del grupo de usuarios, con lo que surgen las computadoras especializadas para determinados tipos de tareas.

### 1.4.4 VENTAJAS

- Funcionamiento autónomo de los sistemas locales, lo que origina un buen tiempo de respuesta.
- Los sistemas de información llegan a todos los departamentos de la empresa.

### 1.4.5 DESVENTAJAS

- Requiere un intenso flujo de informaciones (muchas veces no útiles, como pantallas y datos incorrectos) dentro de la red, lo que puede elevar los costos de comunicaciones.
- Supone una mayor complejidad.
- Si los sistemas no están integrados, pueden producirse problemas de inconsistencia de datos.

## 1.5 MODELO BASADO EN WEB

Nace debido a la expansión de las organizaciones, las empresas cuentan con un número importante de sucursales, repartidas a lo largo de su país y en muchos casos en otros países y continentes. Con estos nuevos retos fue necesario incrementar el rendimiento de los

modelos distribuidos y recurrir a controles centralizados. El nacimiento del modelo basado en Web permite utilizar los esquemas anteriores para su función interna y presentar un marco de comunicación fácil de usar. Las funciones se encuentran en un servidor central el cual a su vez puede o no depender o concentrar a otros, esta característica permite que sólo los usuarios autorizados a ciertas zonas accedan a estos sistemas, es así que un contador en China no podrá acceder a la auditoria de la filial en Brasil si no es autorizado por esta última filial o por la controladora.

Un ambiente basado en Web ofrece una rápida comunicación entre los servidores y los clientes, permitiendo la obtención de la información que se encuentra en los servidores, hace unos años el usuario carecía de poder de procesamiento del lado de los servidores como el ofrecido en otros modelos, problema que con el paso del tiempo se ha ido superando con la aparición de las siguientes generaciones de servidores y de las mejoras incluidas en los lenguajes de programación. Las características generales que se requieren para brindar y soportar un ambiente basado en Web, se listan a continuación -Ortal Robert-.

### 1.5.1 CARACTERISTICAS

- **Servicio**: La primera generación de sistemas basados en Web, ofrecía solamente la capacidad de presentar a los usuarios el contenido predefinido por la empresa, aun cuando se pensaba que era una desventaja, estos modelos fueron bien recibidos. Con la llegada de la segunda generación de lenguajes y servidores, se presentó la oportunidad de ejecutar código del lado del servidor incrementando el valor agregado a la información que estos sistemas almacenaban.
- **Recursos compartidos:** Un servidor Web puede atender cientos de clientes por minuto, de forma concurrente y sin interferir con las acciones de otros usuarios. Los servicios ofrecidos han ido en aumento con el paso de los años y otros servicios como la compartición de archivos, el almacenaje remoto y la impresión vía Internet se han vuelto más populares.
- **Protocolos:** Con protocolos de probada eficiencia como son TCP/IP, IPX, X25 entre otros ha sido posible establecer la comunicación con varios clientes, los cuales inician las peticiones de servicios o recursos mediante reglas de control definidas por la empresa.
- **Transparencia de localización:** El servidor es un proceso que se ejecuta o reside en uno o más equipos de cómputo dentro de un ambiente de red, el cual es localizado por una dirección única.
- **Escalabilidad:** La escalabilidad requiere de un análisis detallado, son varios los tipos de efectos que provoca una baja en la calidad del servicio. Las granjas de servidores pueden aceptar diferentes tipos de hardware y su crecimiento implica un consumo del ancho de banda disponible en el entorno de los servidores, los servidores deben estar siempre activos, lo que repercute en una mayor disponibilidad de nuestra aplicación; el incremento en el ancho de banda, se verá reflejado en la grata experiencia para nuestro personal o clientes.
- **Integridad:** Debido a que toma características del modelo distribuido la integridad del contenido e información se encuentran protegidos de problemas de redundancia, pérdidas y permiten mantener una aplicación en uso.

### 1.5.1.1 CARACTERÍSTICAS FÍSICAS

- Una o más bases de datos robustas, donde residen todos los datos.
- Los servidores pueden no estar concentrados en una misma área, mantienen conexión por medio de una red de comunicaciones.
- Cada servidor sobre la red tiene la capacidad de cubrir las necesidades de los usuarios locales de forma independiente a los demás equipos.
- Toma importancia la red de comunicación de datos.

### 1.5.1.2 CARACTERÍSTICAS LÓGICAS

- Ejecución de todos los procesos de la zona en el servidor central de la región.
- La dispersión geográfica no es importante, los usuarios se conectan de cualquier parte del mundo y sólo es importante la pertenencia del usuario a los sitios que desea consultar.

## 1.5.2 VENTAJAS

- Equipos cliente de bajo costo, independientes de la tecnología usada en los servidores.
- Independencia del usuario, el cual ya no se encuentra atado a su equipo o red de trabajo.

## 1.5.3 DESVENTAJAS

- Requiere un intenso flujo de informaciones dentro de la red, lo que puede elevar los costos de comunicaciones.
- Dependencia forzosa con los proveedores de la red de comunicaciones.
- Fallas persistentes durante las horas de mayor tráfico tanto en la red local como exterior.

En la tabla, se muestra la síntesis de las características básicas de cada modelo, elaborada con la visión de los usuarios, propietarios y desarrolladores, exponiendo los beneficios y carencias de cada modelo.

**Tabla 1-1 Ventajas y desventajas de los modelos de aplicación.**

| Modelo | Usuario | Propietarios | Desarrollo |
|---|---|---|---|
| Centralizado | Pros: Arranque instantáneo, comportamiento predecible, poca huella. | Pros: Control total, bajo costo de equipos de escritorio, altamente seguro, bajo soporte al usuario final. | Pros: Entorno de cliente predecible, herramientas maduras. |
| | Contras: Tamaño delimitado, sólo texto, sólo teclado. | Contras: Alto costo de host, aplicaciones limitadas y costosa, alto costo de | Contras: Muy por detrás de los estándares modernos, poco espacio para la creatividad, problemas multiusuario. |

| | | | |
|---|---|---|---|
| | | mantenimiento, fácil acceso remoto. | |
| Distribuido | Pros: Libertad, presentación gráfica, muchas aplicaciones. | Pros: Más aplicaciones, menor costo. | Pros: Nuevas tecnologías, acercamiento al usuario, herramientas gráficas. |
| | Contras: Más tiempo de descarga, "envidia de PC", mas distracciones. | Contras: Menos controlado y seguro, esperanza de propiedad de los usuarios, Costo elevado de equipamiento y soporte al usuario final, difícil acceso remoto a las aplicaciones. | Contras: Entorno de cliente impredecible, dificultad de distribución, API complejas. |
| Basado en la Web | Pros: Interfaces gráficas muy ricas, fácil acceso a las aplicaciones, texto bien formateado. | Pros: Control estricto, bajo costo de equipo de usuario, fácil gestión, fácil acceso remoto. | Pros: Tecnologías interesantes, entorno creativo, grandes herramientas, distribución sin esfuerzo. |
| | Contras: Puede ser más lento, menos libertad | Contras: Altos costos de servidores, pocas aplicaciones comerciales | Contras: Alta curva de aprendizaje, API y estándar en evolución y competencia. |

*Yager, 2001*

## 1.6 BASES DE DATOS

Las bases de datos fueron desarrollados a mediados de la década de los cincuentas, siendo una de las principales herramientas que las primeras computadoras ofrecían. Surgieron como extensiones de programas desarrollados en *Fortran* que permitían acceso compartido a los datos. A finales de esta década se desarrollaron métodos de acceso soportados por el sistema operativo (acceso directo y secuencial) y maduraron con los sistemas operativos de segunda y tercera generación (principios de los años sesentas). En esta época se desarrollaron las bases de datos estructuradas jerárquicamente y tiempo después las bases de datos de red. A finales de los años sesentas, *Ted Codd*, investigador de IBM, desarrolló un lenguaje de programación de propósito general que denominó "programación relacional", basado en la teoría de conjuntos y lógica matemática y que contenía el principio de lo que hoy son las bases de datos relacionales.

El objetivo primario de una base de datos es, como su nombre indica, almacenar grandes cantidades de datos organizados siguiendo un determinado esquema o "modelo de datos" que facilite su almacenamiento, recuperación y modificación.

Una base de datos cuenta con su correspondiente gestor para simplificar la administración de las tareas comunes de mantenimiento: el sistema manejador de bases de datos (*DBMS:*

*Database Management System*, -Sistema Manejador de bases de datos-). Los *DBMSs* actuales se encuentran perfectamente estandarizados, ofreciendo un número de características y metodologías comunes que posibilitan la comunicación entre diversos tipos y productos comerciales, las bases de datos, cuentan con los mecanismos de conexión necesarios para poder ofrecer su información a muy distintos tipos de lenguajes de programación.

### 1.6.1 CONCEPTOS BÁSICOS DE UN SISTEMA MANEJADOR DE BASES DE DATOS

Todos los conceptos referentes a las bases de datos están hoy muy claros y definidos formalmente. La tecnología de gestión de bases de datos se halla en una etapa muy madura. Las bases de datos han evolucionado durante los pasados treinta años desde sistemas de archivos rudimentarios hasta sistemas gestores de complejas estructuras de datos que ofrecen un gran número de posibilidades. Los principales objetivos de un *DBMS* son los siguientes:

- **Independencia lógica y física de los datos:** se refiere a la capacidad de modificar una definición de esquema en un nivel de la arquitectura sin que esta modificación afecte al nivel inmediatamente superior. Para ello un registro externo en un esquema externo no tiene por qué ser igual a su registro correspondiente en el esquema conceptual.
- **Redundancia mínima:** se trata de usar la base de datos como depósito común de datos para distintas aplicaciones.
- **Acceso concurrente por parte de múltiples usuarios:** control de concurrencia mediante técnicas de bloqueo de datos accedidos.
- **Distribución espacial de los datos:** la independencia lógica y física facilita la posibilidad de sistemas de bases de datos distribuidas. Los datos pueden encontrarse en otra habitación, otro edificio e incluso otro país. El usuario no tiene por qué preocuparse de la localización espacial de los datos a los que accede.
- **Integridad de los datos:** se refiere a las medidas de seguridad que impiden que se introduzcan datos erróneos. Esto puede suceder tanto por motivos físicos (defectos de hardware, actualización incompleta debido a causas externas), como de operación (introducción de datos incoherentes).
- **Consultas complejas optimizadas:** la optimización de consultas permite la rápida ejecución de las mismas.
- **Seguridad de acceso y auditoria:** se refiere al derecho de acceso a los datos contenidos en la base de datos por parte de personas y organismos. El sistema de auditoria mantiene el control de acceso a la base de datos, con el objeto de saber qué o quién realizó una determinada modificación y en qué momento.
- **Respaldo y recuperación:** se refiere a la capacidad de un sistema de base de datos de recuperar su estado en un momento previo a la pérdida de datos.
- **Acceso a través de lenguajes de programación estándar:** se refiere a la posibilidad ya mencionada de acceder a los datos de una base de datos mediante lenguajes de programación ajenos al sistema de base de datos propiamente dicho.

Una base de datos típica conlleva la existencia de tres tipos de usuario con relación a su diseño, desarrollo y uso:

- El administrador de bases de datos (*DBA: -Database Administrador*, Administrador de la base de datos-): diseña y mantiene la base de datos.
- El desarrollador de aplicaciones (programador): implementa las transacciones e interfaces.
- Los usuarios finales: consultan y editan los datos de la base de datos mediante un lenguaje de consulta de alto nivel.

En general, podemos decir que el propósito de una base de datos es doble; responder a consultas sobre los datos que contiene, y ejecutar transacciones.

Una consulta (*query)* se define como una expresión lógica sobre los objetos y relaciones definidos en el esquema conceptual; el resultado es la identificación de un subconjunto lógico de la base de datos. Una transacción consiste en un número de consultas y operaciones de modificación o actualización sobre un subesquema. Las transacciones son atómicas por definición: todos los pasos de una transacción han de ser debidamente ejecutados y confirmados como requisito previo para que la transacción pueda ser llevada a cabo en su conjunto, en caso contrario ha de ser invalidada.

Para llevar a cabo estas tareas, el *DBA* tiene a su disposición la principal herramienta de una base de datos, el sistema gestor de bases de datos -*DBMS*-. A través de éste se realizan todas las operaciones con los datos (consultas y transacciones), de forma que al *DBA* no le atañe la manera en que los datos se encuentran almacenados físicamente, pudiéndose concentrar en los aspectos conceptuales en cuanto a diseño, desarrollo y mantenimiento. Un *DBMS* típico integra los siguientes componentes:

- Un lenguaje de definición de datos (*DDL: Data Definition Language*).
- Un lenguaje de manipulación de datos (*DML: Data Manipulation Language*)
- Un lenguaje de consulta (*QL: Query Language*).
- De forma accesoria, pero ya casi obligada, los *DBMS* modernos añaden un interfaz de usuario gráfico (*GUI: Graphical User Interface*).
- Consultas mediante ejemplo -Interfaces gráficas- (*GQBE: Graphical Query By Example*)

El lenguaje de consulta por excelencia es el llamado *Structured Query Lenguaje (SQL),* que aun con muchas modificaciones y adiciones es un estándar de las *DBMS* relacionales (*RDBMS: Relational Database Management System* –Sistema Manejador de Bases de Datos Relacionales-). Hoy en día, sin embargo, con la llegada de las *DBMS* orientadas a objetos (*ODBMS: Object Database Management System* –Sistemas Manejadores de Bases de Datos Orientadas a Objetos-), otros estándar de consulta se han hecho necesarios; así ha nacido otro estándar, *OQL (Object Query Language* –Lenguaje de consultas orientado a objetos-), como resultado de una de las primeras implementaciones de *ODBMSs* (*O2, de O2 Technologies*). Además, una base de datos puede ser consultada y modificada mediante técnicas "externas", es decir, mediante lenguajes de programación de propósito general, típicamente de tercera generación. Hoy en día, estas técnicas se hallan muy avanzadas,

existiendo estándares que simplifican el acceso a diferentes *DBMSs* de forma transparente, tales como *ODBC* (*Open Database Connectivity*-Conectividad Abierta de Bases de Datos-), que garantizan el acceso a los datos de bases, posiblemente remotas, de distintas compañías.

Por lo que a la representación de información léxica se refiere, los sistemas de bases de datos tradicionales presentan serios problemas. En general, las bases de datos no fueron pensadas para almacenar información compleja, sino grandes cantidades de información relativamente simples, los datos contenidos en una base de datos han de ser por definición atómicos. Esto es necesario para un correcto tratamiento de los mismos, pero por otra parte entorpece la visión de conjunto, esto es, dificulta el tratamiento "inteligente" de entidades complejas.

En una base de datos relacional, las filas representan registros (conjuntos de datos acerca de elementos separados) y las columnas representan campos (atributos particulares de un registro). Al realizar las búsquedas, una base de datos relacional hace coincidir la información de un campo de una tabla con información en el campo correspondiente de otra tabla y con ello produce una tercera tabla que combina los datos solicitados de ambas tablas, una base de datos relacional utiliza los valores coincidentes de dos tablas para relacionar información de ambas.

## 1.7 MODELO DE DATOS CLÁSICO.

### 1.7.1 MODELO JERÁRQUICO

Un *DBMS* jerárquico utiliza jerarquías o árboles para la representación lógica de los datos. Los archivos son organizados en jerarquías, y normalmente cada uno de ellos se corresponde con una de las entidades de la base de datos. Los árboles jerárquicos se representan de forma invertida, con la raíz hacia arriba y las hojas hacia abajo Figura 1-5.

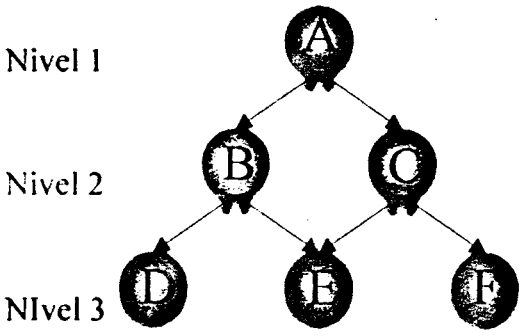



**Figura 1-5 Estructura de un árbol jerárquico.**

Un *DBMS* jerárquico recorre los distintos nodos de un árbol en un preorden que requiere tres pasos:

- Visitar la raíz.
- Visitar el hijo más a la izquierda, si lo hubiera, que no haya sido visitado.
- Si todos los descendientes del segmento considerado se han visitado, volver a su padre e ir al punto 1.

Cada nodo del árbol representa un tipo de registro conceptual, es decir, una entidad. A su vez, cada registro o segmento está constituido por un número de campos que los describen las propiedades o atributos de las entidades. Las relaciones entre entidades están representadas por las ramas.

### 1.7.2 MODELO DE RED

Este modelo fue el resultado de estandarización del comité *CODASYL* (*Conference on Data System Languages* –Conferencia en Sistemas y Lenguajes de Datos-). Aunque existen algunos *DBMSs* de red que no siguen las especificaciones *CODASYL*, en general, una base de datos *CODASYL* es sinónimo de base de datos de red. El modelo de red intenta superar las deficiencias del enfoque jerárquico, permitiendo el tipo de relaciones de muchos a muchos.

Una estructura de datos en red, o estructura plex, es muy similar a una estructura jerárquica, de hecho no es más que un superconjunto de ésta. Al igual que en la estructura jerárquica, cada nodo puede tener varios hijos pero, a diferencia de ésta, también puede tener varios padres. La Figura 1-6 muestra una disposición plex. En esta representación, los nodos C y F tienen dos padres, mientras que los nodos D, E, G y H tienen sólo uno.

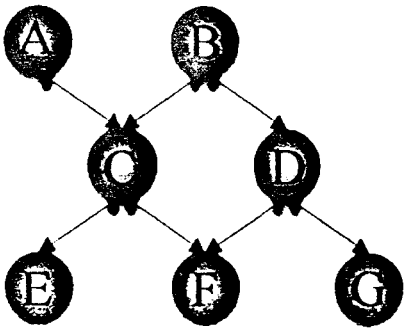



**Figura 1-6 Estructura de datos de red**

El concepto básico en el enfoque de red es el conjunto ('*set*'), definido por el comité *CODASYL*. Un conjunto está constituido por dos tipos de registros que mantienen una relación de muchos a muchos. Para conseguir representar este tipo de relación es necesario que los dos tipos de registros estén interconectados por medio de un registro conector llamado conjunto conector. Los conjuntos poseen las siguientes características:

- El registro padre se denomina propietario del conjunto, mientras que el registro hijo se denomina miembro.
- Un conjunto está formado en un solo registro propietario y uno o más registros miembros.
- Una ocurrencia de conjuntos es una colección de registros, uno de ellos es el propietario y los otros los miembros.
- Todos los registros propietarios de ocurrencias del mismo tipo de conjunto deben ser del mismo tipo de registro.
- El tipo de registro propietario de un tipo de conjunto debe ser distinto de los tipos del registro miembro.
- Sólo se permite que un registro miembro aparezca una vez en las ocurrencias de conjuntos del mismo tipo.
- Un registro miembro puede asociarse con más de un propietario, es decir, puede pertenecer al mismo tiempo a dos o más tipos de conjuntos distintos. Esta situación se puede representar por medio de una estructura multianillo.
- Se pueden definir niveles múltiples de jerarquías donde un tipo de registro puede ser miembro en un conjunto y al mismo tiempo propietario en otro conjunto diferente.

### 1.7.3 MODELO RELACIONAL

El modelo relacional de datos supuso un gran avance con respecto a los modelos anteriores. Este modelo está basado en el concepto de relación. Una relación es un conjunto de n-tuplas. Una tupla, al contrario que un segmento, puede representar tanto entidades como interrelaciones N:M. Los lenguajes matemáticos sobre los que se asienta el modelo relacional, el álgebra y el cálculo relacionales, aportan un sistema de acceso y consultas orientado al conjunto. La repercusión del modelo en los *DBMSs* comerciales actuales ha sido enorme, estando hoy en día la gran mayoría de los gestores de bases de datos basados en mayor o menor medida en el modelo relacional.

El concepto de modelo de datos en sí surgió al mismo tiempo que el modelo relacional de datos fuera propuesto por su creador, *Ted Codd*, después de que los modelos jerárquico y de red estuvieran en uso. Posteriormente, estos dos modelos fueron definidos independientemente de los lenguajes y sistemas usados para implementarlos. Con anterioridad no eran más que colecciones de estructuras de datos y lenguajes sin una teoría subyacente definida. En cuanto al modelo relacional, no se puede decir que sea en sí un modelo semántico de datos. Su enorme éxito no se debe a que permita de forma implícita operaciones conceptualmente abstractas sobre los datos, sino a los altos niveles de fiabilidad e integridad que aporta en el manejo de grandes cantidades de datos.

Desde su comienzo en 1970 y durante mucho tiempo después, los sistemas gestores de bases de datos relacionales (*RDBMS*: *Relational Database Management System* –Sistemas Manejadores de Bases de Datos Relacionales-) estuvieron restringidos al ámbito de los *mainframes* y estaciones de trabajo. Con la irrupción masiva en el mercado de las computadoras personales, aparecieron algunas implementaciones de *RDBMSs* que intentaban emular las propiedades de los grandes sistemas, aunque no contaban con la mayor parte de las características necesarias para ser denominados "relacionales",

especialmente en lo que se refiere al cumplimiento de las reglas de integridad relacional. Hoy en día contamos con *RDBMSs* para computadoras personales que sí pueden ser considerados plenamente relacionales y que, si bien no llegan alcanzar las prestaciones de los grandes sistemas en cuanto a velocidad de ejecución, seguridad, integridad de datos, recuperación y estabilidad, no tienen nada que envidiar a éstos cualitativamente, y sus deficiencias se deben sobre todo al tipo de equipo en el que funcionan y a los sistemas operativos que estas máquinas utilizan.

Lo que realmente marca la diferencia entre los sistemas relacionales y los sistemas anteriores es el hecho de que su creador, *Ted Codd*, basó expresamente su funcionamiento sobre un modelo matemático muy específico: el álgebra relacional y el cálculo relacional, así como la progresiva adopción, por parte de su creador y algunos colaboradores, de un número de Reglas de Integridad Relacional y de Formas Normales.

## 1.8 PARADIGMA DE PROGRAMACIÓN

### 1.8.1 PARADIGMAS DE LOS LENGUAJES DE PROGRAMACIÓN

Un paradigma es una forma o manera de ver las cosas. En los lenguajes de programación actuales existen cuatro modelos que los describen, estos son:

- Lenguajes imperativos o de procedimientos
- Lenguajes aplicativos o funcionales
- Lenguajes con base en reglas o lógico
- Lenguajes orientados a objetos
- Lenguajes concurrentes, paralelos y distribuidos

### 1.8.2 LENGUAJES IMPERATIVOS O DE PROCEDIMIENTOS

Estos derivan su nombre del papel que juegan los enunciados imperativos que se combinan entre si para alcanzar un resultado deseado en un programa. Este tipo de programas se componen de una serie de enunciados, y la ejecución de cada enunciado hace que el intérprete cambie el valor de una localidad o más en su memoria, es decir que pase a un nuevo estado. La sintaxis de esta clase de lenguajes tiene por lo general la forma:

*Enunciado 1*

*Enunciado 2*

Este tipo de lenguajes se adhieren a la arquitectura convencional de una computadora, que realiza operaciones de manera secuencial. Ejemplos de este tipo de lenguajes son; *Algol 60*, *Pascal*, *C*, *Cobol*, *Fortran*, entre otros.

El punto problemático de los lenguajes imperativos es la imposibilidad que existe de demostrar que los programas estén correctos. Esta dificultad es causada por el hecho de que lo correcto de un programa depende del contenido de todas y cada una de las celdas de

memoria. Para poder observar un programa a través del tiempo debemos tomar "fotos instantáneas" de la memoria antes y después de ejecutar cada paso. Si el programa maneja una cantidad grande de memoria esto se vuelve tedioso en el mejor de los casos y prácticamente imposible en general.

### 1.8.3 LENGUAJES APLICATIVOS O FUNCIONALES

Este tipo de lenguajes consisten en examinar la función que el programa representa, y no sólo los cambios de estado conforme el programa se ejecuta, enunciado por enunciado. Esto se puede conseguir observando el resultado deseado en vez de los datos disponibles. En otras palabras, en vez de examinar la serie de estados a través de los cuales debe pasar la máquina para obtener una respuesta, la pregunta que debe formularse es: ¿Cuál es la función que se debe aplicar al estado de máquina inicial accediendo al conjunto inicial de variables y combinándolas en forma especifica para obtener una respuesta? Los lenguajes que hacen énfasis en este punto de vista son los lenguajes funcionales.

Las características generales de este tipo de lenguajes son:

- Un conjunto de funciones primitivas: Son el conjunto de funciones predefinidas en el lenguaje que pueden ser aplicadas.
- Un conjunto de formas funcionales: Son los mecanismos mediante los cuales podemos combinar funciones para crear funciones nuevas.
- Las operaciones de aplicación: Es el mecanismo construido en el lenguaje para aplicar una función a sus argumentos y obtener un valor.
- Un conjunto de objetos de datos: Son los objetos permitidos del dominio y el rango.

Generalmente los lenguajes funcionales están muy restringidos en cuanto a la variedad de objetos de datos que permiten, siendo éstos conjuntos de una estructura simple y regular.

En general la sintaxis de estos lenguajes es similar a:

*funciónn(...función2(función1(datos))...)*

Ejemplos: *LISP, ML*

Los matemáticos desde hace un buen tiempo están resolviendo problemas usando el concepto de función. Una función convierte ciertos datos en resultados. Si supiéramos cómo evaluar una función, usando la computadora, podríamos resolver automáticamente muchos problemas. Así pensaron algunos matemáticos, que no le tenían miedo a la máquina, e inventaron los lenguajes de programación funcionales. Además, aprovecharon la posibilidad que tienen las funciones para manipular datos simbólicos, y no solamente numéricos, y la propiedad de las funciones que les permite componer, creando de esta manera, la oportunidad para resolver problemas complejos a partir de las soluciones a otros más sencillos. También se incluyó la posibilidad de definir funciones recursivamente.

Un lenguaje funcional ofrece conceptos que son muy entendibles y relativamente fáciles de manejar para todos los que no se durmieron en las clases de matemáticas. El lenguaje

funcional más antiguo, y seguramente el más popular hasta la fecha, es *LISP*. Su área de aplicación es principalmente la Inteligencia Artificial. En la década de los 80 hubo un nuevo interés por los lenguajes funcionales, añadiendo la tipificación y algunos conceptos modernos de módulos y polimorfismo, como es el caso del lenguaje *ML*.

Programar en un lenguaje funcional significa construir funciones a partir de las ya existentes. Por lo tanto es importante conocer y comprender bien las funciones que conforman la base del lenguaje, así como las que ya fueron definidas previamente. De esta manera se pueden ir construyendo aplicaciones cada vez más complejas. La desventaja de este modelo es que resulta bastante alejado del modelo de la máquina de *Von Neumann* y, por lo tanto, la eficiencia de ejecución de los intérpretes de lenguajes funcionales no es comparable con la ejecución de los programas imperativos precompilados. Para remediar la deficiencia, se está buscando utilizar arquitecturas paralelas que mejoren el desempeño de los programas funcionales, sin que hasta la fecha estos intentos tengan un impacto real importante.

### 1.8.4 LENGUAJES CON BASE EN REGLAS O LÓGICO

Este tipo de lenguajes se ejecutan verificando una condición habilitadora, y cuando se satisface, ejecutan una acción. Las condiciones habilitadoras determinan el orden de ejecución, la sintaxis de esta clase de lenguajes es de la forma:

*condición1 entonces acción1*

*condición2 entonces acción2*

...

*Condición(n) entonces acción(n)*

Ejemplos: *PROLOG*

El conocimiento básico de las matemáticas se puede representar en la lógica en forma de axiomas, a los cuales se añaden reglas formales para deducir cosas verdaderas (teoremas) a partir de los axiomas. Gracias al trabajo de algunos matemáticos, de finales de siglo pasado y principios de éste, se encontró la manera de automatizar el razonamiento lógico - particularmente para un subconjunto significativo de la lógica de primer orden- que permitió que la lógica matemática diera origen a otro tipo de lenguajes de programación, conocidos como lenguajes lógicos. También se conoce a estos lenguajes, y a los funcionales, como lenguajes declarativos, porque el programador, para solucionar un problema, todo lo que tiene que hacer es describirlo vía axiomas y reglas de deducción en el caso de la programación lógica y vía funciones en el caso de la programación funcional.

En los lenguajes lógicos se utiliza el formalismo de la lógica para representar el conocimiento sobre un problema y para hacer preguntas que, si se demuestra que se pueden deducir a partir del conocimiento dado en forma de axiomas y de las reglas de deducción estipuladas, se vuelven teoremas. Así se encuentran soluciones a problemas formulados como preguntas. Con base en la información expresada dentro de la lógica de primer orden,

se formulan las preguntas sobre el dominio del problema y el intérprete del lenguaje lógico trata de encontrar la respuesta automáticamente. El conocimiento sobre el problema se expresa en forma de predicados (axiomas) que establecen relaciones sobre los símbolos que representan los datos del dominio del problema.

El *PROLOG* es el primer lenguaje lógico y el más conocido y utilizado. Sus orígenes se remontan a los inicios de la década de los 70. También en este caso, las aplicaciones a la Inteligencia Artificial mantienen el lenguaje vivo y útil.

En el caso de la programación lógica, el trabajo del programador se restringe a la buena descripción del problema en forma de hechos y reglas. A partir de ésta se pueden encontrar muchas soluciones dependiendo de como se formulen las preguntas (metas), que tienen sentido para el problema. Si el programa está bien definido, el sistema encuentra automáticamente las respuestas a las preguntas formuladas. En este caso ya no es necesario definir el algoritmo de solución, como en la programación imperativa, en cambio, lo fundamental aquí es expresar bien el conocimiento sobre el problema mismo. En programación lógica, al igual que en programación funcional, el programa, en este caso los hechos y las reglas, están muy alejados del modelo *Von Neumann* que posee la máquina en la que tienen que ser interpretados; por lo tanto, la eficiencia de la ejecución no puede ser comparable con la de un programa equivalente escrito en un lenguaje imperativo. Sin embargo, para cierto tipo de problemas, la formulación del programa mismo puede ser mucho más sencilla y natural (para un programador experimentado, por supuesto).

### 1.8.5 LENGUAJES ORIENTADOS A OBJETOS

Un lenguaje orientado a objetos es aquél que utiliza y maneja entes (objetos) a través de sus propiedades y características de objetos o entes reales.

A mediados de los años 60 se empezó a vislumbrar el uso de las computadoras para la simulación de problemas del mundo real. Pero el mundo real está lleno de objetos, en la mayoría de los casos complejos, los cuales difícilmente se traducen a los tipos de datos primitivos de los lenguajes imperativos. El concepto de objeto y sus colecciones, llamadas clases de objetos, permitieron introducir abstracciones de datos a los lenguajes de programación. La posibilidad de reutilización del código y sus indispensables modificaciones, se reflejaron en la idea de las jerarquías de herencia de clases, el concepto de polimorfismo fue introducido vía procedimientos virtuales. Todos estos conceptos fueron presentados en el lenguaje Simula 67, desde el año 1967. Aunque pensado como lenguaje de propósito general, Simula tuvo su mayor éxito en las aplicaciones de simulación discreta, gracias a la clase *SIMULATION* que facilitaba considerablemente la programación.

La comunidad informática ha tardado demasiado en entender la utilidad de los conceptos básicos de *Simula 67*, que hoy identificamos como conceptos del modelo de objetos. Tuvimos que esperar hasta la década de 1980 para vivir una verdadera ola de propuestas de lenguajes de programación con conceptos de objetos encabezada por *Smalltalk*, *C++*, *Eiffel*, *Modula-3*, *Ada 95* y terminando con *Java*. La moda de objetos se ha extendido de los lenguajes de programación a la Ingeniería de Software.

El modelo de objetos, y los lenguajes que lo usan, facilitan la construcción de sistemas o programas en forma modular. Los objetos ayudan a expresar programas en términos de abstracciones del mundo real, lo que aumenta su comprensión. La clase ofrece cierto tipo de empleo de módulos que facilitan las modificaciones al sistema. La reutilización de clases previamente probadas en distintos sistemas también es otro punto a favor. Sin embargo, el modelo de objetos, a la hora de ser interpretado en la arquitectura *Von Neumann* conlleva un excesivo manejo dinámico de memoria debido a la constante creación de objetos, así como a una carga de código fuente causada por la constante invocación de métodos. Por lo tanto, los programas en lenguajes orientados a objetos siempre pierden en eficiencia, en tiempo y memoria, contra los programas equivalentes en lenguajes imperativos.

### 1.8.6 LENGUAJES CONCURRENTES, PARALELOS Y DISTRIBUIDOS

La necesidad de ofrecer concurrencia en el acceso a los recursos computacionales se remonta a los primeros sistemas operativos. Mientras que un programa realizaba una operación de entrada o salida otro podría gozar del tiempo del procesador para sumar dos números, por ejemplo. Aprovechar al máximo los recursos computacionales fue una necesidad apremiante, sobre todo en la época en que las computadoras eran caras y escasas; el sistema operativo tenía que ofrecer la ejecución concurrente y segura de programas de varios usuarios, que desde distintas terminales utilizaban un solo procesador, y así surgió la necesidad de introducir algunos conceptos de programación concurrente para programar los sistemas operativos.

Posteriormente, cuando los procesadores cambiaron de tamaño y de precio, se abrió la posibilidad de contar con varios procesadores en una máquina y ofrecer el procesamiento en paralelo, es decir, procesar varios programas al mismo tiempo. Esto dio el impulso a la creación de lenguajes que permitían expresar el paralelismo. Finalmente, llegaron las redes de computadoras, que también ofrecen la posibilidad de ejecución en paralelo, pero con procesadores distantes, lo cual conocemos como la programación distribuida.

En resumen, el origen de los conceptos para el manejo de concurrencia, paralelismo y distribución está en el deseo de aprovechar al máximo la arquitectura *Von Neumann* y sus modalidades reflejadas en conexiones paralelas y distribuidas.

Históricamente encontramos en la literatura soluciones conceptuales y mecanismos tales como: semáforos, regiones críticas, monitores, envío de mensajes (*CSP*), llamadas a procedimientos remotos (*RPC*), que posteriormente se incluyeron como partes de los lenguajes de programación en *Concurrent Pascal*, *Modula*, *Ada*, *OCCAM*, y últimamente en *Java*.

Uno de los ejemplos más importantes es el modelo de envío de mensajes de *CSP*, para las arquitecturas paralelas y distribuidas, el cual no solamente fructificó en una propuesta del lenguaje de programación *OCCAM*, sino dio origen a una nueva familia de procesadores, llamados "*transputers*", que básicamente se componen de una arquitectura paralela.

Es difícil evaluar las propuestas existentes de lenguajes para la programación concurrente, paralela y distribuida. Primero, porque los programadores están acostumbrados a la

programación secuencial y cualquier uso de estos mecanismos les dificulta la construcción y el análisis de programas. Por otro lado, este tipo de conceptos en el pasado fue manejado principalmente a nivel de sistemas operativos, protocolos de comunicación, entre otros, donde la eficiencia era crucial, y por lo tanto no se utilizaban lenguajes de alto nivel para la programación. Hoy en día, la programación de sistemas complejos tiene que incluir las partes de comunicaciones, programación distribuida y concurrencia. Esto lo saben los creadores de los lenguajes más recientes, que integran conceptos para manejar: los hilos de control, comunicación, sincronización y no determinismo; el hardware y las aplicaciones se los exigen.

## 1.9 ARQUITECTURA

En la década de los ochenta las organizaciones tenían que elegir entre computadoras personales o grandes *Mainframes* para su procesamiento de datos. Las computadoras personales, aunque económicas y cada vez más poderosas, carecían de los servicios que un *Mainframe* podía ofrecer; uno de estos servicios fue compartir información de manera instantánea entre los diferentes usuarios del sistema. Para hacer que las computadoras personales compartieran información se crearon las redes, a las que también se integraron los *mainframes*. A finales de la década de los ochentas se introdujo el término "cliente/servidor".

### 1.9.1 ARQUITECTURA CLIENTE SERVIDOR

La arquitectura cliente/servidor promete facilidad de uso, flexibilidad, interoperabilidad y escalabilidad. En las aplicaciones Web se utiliza desde las aplicaciones de dos capas en adelante.

### 1.9.2 ARQUITECTURAS DE DOS CAPAS

Las arquitecturas de dos capas consisten de tres componentes distribuidos en dos capas: cliente (solicitante de servicios) y servidor (proveedor de servicios). Los tres componentes son:

- **Interfaz de usuario al sistema**. Tales como una sesión, entradas de texto, desplegado de menús, entre otros.
- **Administración de procesamiento**. Tales como la ejecución de procesos, el monitoreo de los mismos y servicios de procesamiento de recursos.
- **Administración de bases de datos**. Tales como los servicios de acceso a datos y archivos.

El diseño de dos capas coloca la interfaz de usuario exclusivamente en el cliente. Coloca la administración de base de datos en el servidor y divide la administración de procesos entre el cliente y/o el servidor, creando únicamente dos capas. En esta interacción los clientes se pueden dividir en clientes gruesos y clientes delgados.

### 1.9.3 CLIENTE GRUESO

Un cliente grueso, algunas veces conocido como cliente servidor de dos niveles, es cuando una interfaz muy completa se comunica con una base de datos de soporte. Una aplicación tal como *Microsoft Access* puede ejecutarse en nuestra computadora personal y comunicarse con un servidor, el cual contiene nuestra base de datos de *SQL Server 2000.* Esta interfaz completa tiene integradas muchas de las capas lógicas de la arquitectura cliente/servidor, como presentación, reglas de negocios e incluso algo de la capa de datos.

Como este tipo de aplicación tiene la mayor parte de la lógica contenida en la aplicación nos permite verificar la entrada de datos y devolver los mensajes de error con mucha rapidez, evitando de esta manera un viaje de ida y vuelta al servidor.

Pero, ¿cuáles son los beneficios reales de este tipo de arquitectura? La arquitectura de cliente grueso nos permite:

#### 1.9.3.1 BENEFICIOS

- Validar los datos con mucha rapidez.
- Administrar la seguridad más fácilmente.
- Por lo general realizar el desarrollo fácil y rápidamente.
- Reducir el tráfico en la red.
- Aprovechar los componentes del sistema operativo.
- Utilizar servidores de bajo costo.
- Si el servidor falla, la aplicación local aún podría estar disponible.

#### 1.9.3.2 DESVENTAJAS

- Debe volver a implementarse en cada computadora personal, en el caso de requerir una modificación en el código.
- Es muy costosa en computadora personal cliente, debido a que el cliente grueso tiende a utilizar los recursos en forma intensiva.
- Puede ser muy difícil de mantener, especialmente si la aplicación se implementa en muchas computadoras personales cliente.
- Su costo de soporte puede incrementarse considerablemente.
- Estos no son todos los beneficios y desventajas para la arquitectura de cliente grueso, pero nos muestran cual es el ámbito de operación de un cliente grueso.

### 1.9.4 CLIENTE DELGADO

Un cliente delgado, algunas veces conocido como cliente/servidor de múltiples niveles, nos permite obtener una división más definitiva entre las capas de nuestra aplicación. Cliente delgado se refiere más a la ola de aplicaciones recientes basadas en Web que van prevaleciendo cada vez más en todo el mundo-

Con este tipo de arquitectura podemos colocar la presentación en la *PC* cliente, las reglas de negocios en un servidor central y la capa de datos en nuestro *SQL Server.* Esto tiene más sentido que la arquitectura de cliente grueso, ya que cada capa está separada lógicamente de

la otra. Aunque cada capa está separada, cada una depende considerablemente de las otras dos para que la aplicación funcione como se espera.

### 1.9.4.1 BENEFICIOS

- Si se requiere una modificación en el código, sólo necesita hacer el cambio en un solo lugar.
- Es muy fácil de implementar.
- Es fácil de mantener, ya que todo el código se guarda en un lugar central.
- Como la capa de presentación se ejecuta sólo en la *PC* cliente, se necesitan muy pocos recursos.
- Se basa lógicamente en el modelo de la arquitectura cliente/servidor.

### 1.9.4.2 DESVENTAJAS

- Se incrementa el tráfico en la red, ya que cada solicitud requiere de un viaje al servidor.
- Se necesitan servidores costosos para ejecutar y administrar la aplicación.
- Por lo general, no está disponible el acceso al sistema operativo, por lo que perdemos esta funcionalidad.
- La seguridad puede llegar a convertirse en una pesadilla.
- La base de código puede ser muy extensa y difícil de mantener.

Son las necesidades al implementar una aplicación las que definen el tipo de cliente que será usado en nuestra aplicación como ejemplo: Si se tienen 200 usuarios, y todos con especificaciones distintas en sus equipos, incluso algunos antiguos 486, tal vez se necesite considerar la arquitectura de cliente delgado. No obstante, si se va a implementar para 50 usuarios y la red está un poco lenta, pero las computadoras personales tienen altas especificaciones, entonces la arquitectura de cliente grueso podría adaptarse mejor a las necesidades.

Existen varias razones por las cuales se debe escoger una arquitectura en vez de la otra. El conocimiento de los programadores es una razón. Si los programadores no han creado anteriormente un sitio Web, les tomará mucho más tiempo desarrollar una aplicación utilizando la arquitectura de cliente delgado que la de cliente grueso.

Para seleccionar mejor el tipo de arquitectura cliente, en la Tabla 1-2 se presenta un cuadro con la aplicación mas adecuada de estas arquitecturas.

**Tabla 1-2 Escenarios típicos del uso de arquitecturas de cliente-servidor**

| Arquitectura | Aplicaciones |
|---|---|
| Una arquitectura cliente-servidor de dos niveles con clientes delgados | Aplicaciones de sistemas heredados donde es poco práctico separar el procesamiento de la aplicación y la administración de datos.<br>Aplicaciones intensivas como compiladores con poca o ninguna administración de datos<br>Aplicaciones de datos intensivas (navegadores y consultas a una base de datos) con poco o ningún procesamiento de la aplicación. |
| Una arquitectura cliente-servidor | Aplicaciones donde el procesamiento de la aplicación es |

| | |
|---|---|
| de dos niveles con clientes gruesos | proveído por software en el cliente. Aplicaciones donde se requiere procesamiento de datos intensivo (ejemplo, visualización de datos). Aplicaciones con funcionalidad relativamente estable para los usuarios finales, usadas en un ambiente con administración de sistemas establecida. |
| Arquitectura de cliente-servidor de tres niveles o más niveles | Aplicaciones de gran escala con cientos o miles de clientes. Aplicaciones donde tanto los datos como la aplicación son volátiles. Aplicaciones donde los datos de múltiples fuentes son integrados. |

## 1.9.5 ARQUITECTURA DE N CAPAS

La primera forma de concebir nuestros programas fue por medio de un diseño de una Capa o *Single-Tier* aun cuando no sabíamos. Luego llegaron los servidores, y se empezó a trabajar en dos capas o Cliente - Servidor, con el paso del tiempo se llegó al desarrollo en tres capas, (*Three-Tier Development*).

La idea del desarrollo en tres capas consiste en utilizar un método de desarrollo para nuestros sistemas que nos permita separar esto en distintas capas. Las capas recomendadas son: La interfaz de usuario, las reglas de negocio y la Base de Datos, la idea de esta arquitectura estaba basada principalmente en la capacidad de escalabilidad que esto nos ofrece, por ejemplo, si tenemos desarrollada una aplicación basada en un motor de datos de Access, y si en el desarrollo de esta, aplicamos apropiadamente las reglas de diseño de tres capas, cuando quisiéramos llevar la misma a funcionar, por ejemplo, con *SQL Server*, no deberíamos tocar más que el motor de datos, la capa cliente quedaría intacta y, como mucho algún "toque" muy superficial pudiera sufrir la capa de reglas de negocios. La Figura 1-7 nos muestra el concepto del funcionamiento de esta arquitectura.

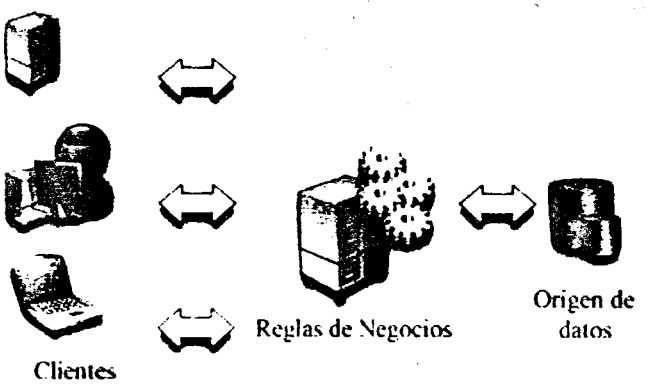



**Figura 1-7 Diseño en tres capas.**

Como lo podemos ver, esta arquitectura nos permite hacer que tanto la interfaz de usuario, las reglas de negocios y el motor de datos se conviertan en entidades separadas unas de otras, lo importante es mantener bien definidas las interfaces que cada una de estas expongan para comunicarse con la otra.

Al día de hoy, ya nos encontramos hablando de desarrollar aplicaciones "n capas", la que más comúnmente tenemos entre nosotros es la de cuatro capas, la capa que se agrega es la que surge de separar definitivamente las reglas de negocio de la de "Acceso a Datos". Esta arquitectura nos brinda la ventaja de aislar definitivamente nuestra lógica de negocios de todo lo que tenga que ver con el origen de datos, ya que desde el manejo de la conexión, hasta la ejecución de una consulta, la manejará la capa de Acceso a Datos. De este modo, ante cualquier eventual cambio, solo se deberá tocar un módulo específico, así como al momento de plantear la escalabilidad de nuestro sistema, si hemos respetado las reglas básicas de diseño no deberíamos afrontar grandes modificaciones. Este concepto lo ejemplifican en la mayoría de los seminarios con la imagen de un edificio con zonas de absorción de sismos, cada sección se encuentra aislada del resto de la estructura y es posible minimizar el impacto que otras zonas reciban (Figura 1-8).

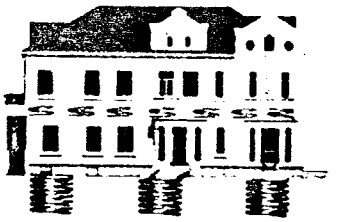


**Figura 1-8 Ejemplo de un diseño multicapas usado en seminarios**

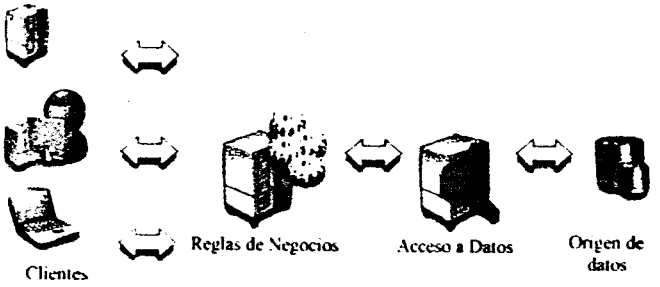



**Figura 1-9 Modelo de diseño en cuatro capas.**

### 1.9.6 IMPORTANCIA DE ESTA ARQUITECTURA

Existen muchas razones por las que usar el desarrollo de aplicaciones bajo la arquitectura de n Capas:

- Abstracción total acerca del origen de datos. Las distintas capas se especializan absolutamente en la funcionalidad que deben brindar (procesamiento en las reglas de negocios o presentación de datos en la capa cliente) sin importar cual es el origen de los datos procesados.
- Bajo costo de desarrollo y mantenimiento de las aplicaciones. Si bien al momento del diseño podemos observar una mayor carga de complejidad, la utilización de esta arquitectura nos brinda un control más cercano de cada componente, así como también la posibilidad de una verdadera reutilización del código. Por ejemplo: Si tuviésemos en nuestras reglas de negocios una función que nos liste el resumen de cuenta de nuestros clientes cuando un usuario de nuestra aplicación, desde su capa cliente ingrese a este módulo se ejecuta MostrarListaCuenta() para ver el detalle en su monitor; pero si el desarrollador del sitio Web de la empresa, al momento de armar la página de consulta nos pide esto, le diremos que simplemente cargue el componente de reglas de negocios y ejecute MostrarListaCuenta(). Como podemos ver, este es un concreto ejemplo de reutilización de código y a su vez, una herramienta que nos ofrece seguridad, ya que mediante la utilización de esta arquitectura, el desarrollador de una interfaz de usuario jamás llegará a manipular directamente un dato en nuestro servidor si no es a través de un componente.
- Estandarización de las reglas de negocio. Las reglas de negocio se encuentran encapsuladas en un *set* de rutinas comunes y pueden ser llamadas desde diversas aplicaciones sin necesidad de saber cómo funciona ésta o ha sido diseñada.

Las aplicaciones en 'n' capas se han convertido en el modelo para el desarrollo de software empresarial actual. Para muchos desarrolladores, una aplicación en 'n' niveles es algo divido en distintas partes lógicas. La opción más habitual está formada por una división en tres partes, las cuales son; presentación, lógica de negocios y datos, existen otras posibilidades de construir esta división. En la Tabla 1-3 se describe brevemente las capas antes mencionadas

**Tabla 1-3 Capas más representativas en una arquitectura de 'n' capas**

| Capa | Funcionalidad |
|---|---|
| Capa de Presentación | Interfaz de usuario. Manipula toda la interacción con el usuario. |
| Capa de Negocio | Lógica empresarial, procesos, fórmulas, la capa "racional" de una aplicación de tres capas. |
| Capa de acceso a datos | Manipula el almacenamiento y recuperación de los datos persistentes. |

*Yager, 2001*

Las aplicaciones de 'n' niveles surgieron como una forma de resolver algunos de los problemas asociados a las aplicaciones cliente/servidor tradicionales, pero debido a la llegada del modelo basado en Web, esta arquitectura ha llegado a dominar el nuevo desarrollo.

Factorizar una aplicación en partes lógicas resulta útil. Dividir software de gran tamaño en partes más pequeñas puede hacerlo más fácil de generar, reutilizar y modificar. También resulta muy útil porque permite acomodar diferentes tecnologías o diferentes organizaciones de negocios.

Un diseño de 'n' capas nos proporciona la flexibilidad necesaria para presentar una solución que pueda con el tiempo seguir respondiendo a las necesidades de nuestro cliente.

# CAPÍTULO 2 ANÁLISIS DE LA INFORMACIÓN

La idea de llevar a un ambiente basado en Web los productos generados por el "PROGRAMA DE MANEJO AMBIENTAL DE LOS RESIDUOS PROVENIENTES DE DESAZOLVE DEL SISTEMA DE DRENAJE DEL DF Y DE LAS PRESAS, ASÍ COMO DE PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES MUNICIPALES", responde a la necesidad de difundir y utilizar de manera extensiva los conocimientos obtenidos, evitando por este medio, el almacenamiento inútil de extensos informes en papel o formato digital. El disponer de una aplicación tan dinámica y fácil de usar como es una página Web, permitirá obtener de forma práctica datos e información y de ser necesario se podrá profundizar en el tema por medio de los informes almacenados por esta aplicación.

Surge una interrogante en cuanto a la selección del modelo de aplicación; "¿el modelo basado en Web, es la solución a las necesidades de uso de nuestro cliente?", esta inquietud nace debido a que nunca se invitó a un analista o se entregó algún estudio previo para determinar la viabilidad del uso de una solución basada en Web. Para despejar esta incógnita y poder orientar correctamente el desarrollo de nuestra solución, es preciso iniciar este capitulo con un breve comentario acerca de los requerimientos técnicos y humanos que serán necesarios para el correcto funcionamiento de nuestra aplicación en su medio ambiente final.

## 2.1 PLANIFICACIÓN PREVIA Y CONSIDERACIONES DE DISEÑO

DESCRIPCIÓN DEL "PROGRAMA DE MANEJO AMBIENTAL DE LOS RESIDUOS PROVENIENTES DE DESAZOLVE DEL SISTEMA DE DRENAJE DEL DF Y DE LAS PRESAS, ASÍ COMO DE PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES MUNICIPALES"

Debido a la naturaleza interdisciplinaria del trabajo original del cual deriva nuestro proyecto de tesis, es necesario presentar un resumen del contexto teórico. Al ser un estudio ambiental, se presenta un resumen compacto, que expone la situación actual de los azolves en el sistema de drenaje del Distrito Federal.

### 2.1.1 RESUMEN

El proyecto original cuantificó la generación de los azolves extraídos del sistema de drenaje de agua residual de la ciudad de México, se determinó la extracción promedio anual y se evaluaron sus características. Se determinó que el origen de los azolves no se debe únicamente a la sedimentación de sólidos contenidos en las aguas residuales, domésticas, pluviales e industriales sino que además son generados por basura arrastrada por el agua de lluvia, el vertido clandestino de desechos sólidos a la infraestructura hidráulica, la descarga del 97% de los lodos generados en las plantas de tratamiento de aguas residuales y el suelo, principalmente proveniente de la zona de conservación, por la erosión pluvial. Con base en su origen los azolves se clasifican en dos grupos: los que provienen de los sedimentos retenidos en presas, lagunas y vasos de regulación de la infraestructura hidráulica y lo que se obtienen a la salida de las estaciones de transferencia de lodos. En términos de cantidad se estima que en total se generan 2.7 $Mm^3$/año, de los cuales se extraen alrededor de una tercera parte por año. Los datos históricos de extracción son muy variables y dependen de diversas circunstancias, pero la que predomina es la disponibilidad financiera para llevar a cabo el trabajo. No hay manera de conocer con exactitud en cuanto contribuye cada fuente a la generación total de azolves, pero la producción anual de los lodos por las PTAR's ("Plantas de Tratamiento de Aguas Residuales") representa el 15% de la generación y 46% de la extracción y el suelo que se pierde por erosión de la zona de conservación debe representar también un porcentaje alto pero no fue posible determinarlo con precisión. Como aporte principal de este trabajo se presenta y describe el diagrama de flujo de la generación de azolves, esta labor recopila tanto información documentada como el conocimiento oral que no había sido escrito.

Para llevar a cabo el proyecto se tuvieron que diseñar una metodología de monitoreo, tipo y parámetros de muestreo y un sistema de clasificación de azolves, para integrar y analizar fácilmente la información. El sistema de clasificación comprende cinco niveles: El primero (nivel I) corresponde a los azolves que sin ningún tipo de tratamiento pueden ser directamente aprovechados en agricultura, el Nivel II requiere para este mismo fin el control de microorganismos patógenos o de BTX's ("cuantificación de compuestos orgánicos aromáticos como Benceno, Tolueno, Etil Benceno y Xileno") mediante sistemas sencillos de tratamiento, el Nivel III implica problema por HTP's ("Hidrocarburos Totales de Petróleo") y por tanto el empleo de métodos avanzados de tratamiento, el Nivel IV

implica contaminación por metales y por ello implica que los azolves deban ir a un relleno sanitario y, finalmente el nivel V es para aquellos azolves considerados residuos peligrosos. Con esta clasificación se construyó la Tabla 2-1 para los azolves. Los límites empleados para diferenciar niveles corresponden a los señalados por la normatividad nacional relacionada.

**Tabla 2-1 Resumen de resultados de clasificación de azolves**

| Sitio | Nivel de clasificación | | | | |
|---|---|---|---|---|---|
| | I | II | III | IV | V |
| Presas | 0 | 1 | 2 | 2 | 0 |
| Lagunas y vasos de regulación | 0 | 2 | 0 | 1 | 1 |
| Cauces | 0 | 0 | 1 | 0 | 1 |
| Estaciones de transferencia | 0 | 0 | 0 | 0 | 3 |

Para los lodos, se siguió un análisis basado en la NOM 004 ECOL, toda vez que éstos NO constituyen azolves en forma directa, sino se transforman en éstos una vez descargados al drenaje, succionados de éste y pasados a través de estaciones de transferencia. De este estudió resultó que todos requieren control de patógenos y que en algunos casos se tienen problemas por metales (arsénico y níquel, principalmente). En este sentido, también algunos de los azolves provenientes de presas presentaron problemas de metales pero por plomo. Los lodos de las PTAR's una vez mezclados con los materiales del drenaje y pasados por las estaciones de transferencia se tornan residuos peligrosos, por ello se debe promover su tratamiento y recuperación como biosólidos en las plantas.

Por los problemas detectados en los azolves y en lodos de metales, "HTP's" y "BTX" se recomienda fuertemente reforzar el programa de pretratamiento de descargas. Asimismo se recomienda evitar la generación de azolves a través de programas que controlen la erosión, eviten el depósito clandestino de desechos sólidos en la infraestructura hidráulica, se promueva la limpieza en general de la ciudad y se traten los lodos de las "PTARS".

Por otra parte, se determinaron las zonas de la ciudad donde es posible reutilizar los azolves con características controladas por Delegación (Cuajimalpa de Morelos, Xochimilco, Tláhuac, Iztapalapa, Milpa Alta, Tlalpan, Álvaro Obregón y Magdalena Contreras), por uso (agrícola, nivelación de suelos, remediación de suelos y mejoramiento de zonas comunes), por tipo de suelo (feozem, andosol y solonchak) por zona de suelo de conservación (agroecológico, agroforestal y forestal) y para Tlalpan, Tláhuac, Magdalena Contreras y la zona de Chalco (fuera del DF) se mencionan sitios precisos de empleo. Se hizo una evaluación preliminar de las cantidades posibles por depositar, en los casos donde se contó con información. En el caso de los lodos se presenta donde emplearlos como biosólidos (Tabla 2-2)

**Tabla 2-2 Plantas de Tratamiento de Aguas Residuales y las Delegaciones con sitios poten[illegible] aceptar los lodos estabilizados**

| Nombre de la planta / Delegación | Delegación receptoras de lodos recomendadas |
|---|---|
| Iztacalco / Iztacalco | Tláhuac o Xochimilco |
| Cerro de la estrella / Iztapalapa | Tláhuac o Xochimilco |
| San Luis Tlaxialtemalco/ Xochimilco | Xochimilco |
| San Lorenzo / Tláhuac | Tláhuac |
| San Pedro Atocpan / Milpa Alta | Xochimilco |
| San Andrés Mixquic / Tláhuac | Tláhuac |
| Coyoacán / Coyoacán | Tláhuac, Xochimilco o Magdalena Contreras |
| Tlatelolco / Cuauhtémoc | Álvaro Obregón |
| Acueducto de Guadalupe / G.A.M. | Álvaro Obregón |
| San Juan de Aragón / G.A.M. | Álvaro Obregón o Tláhuac |
| Cd. Deportiva / Iztacalco | Tláhuac |
| Bosques de las lomas / M. Hgo. | Álvaro Obregón |
| Chapultepec / Miguel Hidalgo | Álvaro Obregón |
| San Juan Ixtayopan / Tláhuac | Tláhuac |
| San Nicolás Tetelco / Tláhuac | Tláhuac |
| Abasolo / Tlalpan | Xochimilco o Magdalena Contreras |
| Parres / Tlalpan | Xochimilco o Magdalena Contreras |
| PEMEX Picacho / Tlalpan | Xochimilco o Magdalena Contreras |
| San Miguel Xicalco / Tlalpan | Xochimilco o Magdalena Contreras |
| Reclusorio sur / Xochimilco | Xochimilco |

Con objeto de poner en marcha acciones para el manejo controlado de los azolves se propone realizar un programa que incluya:

- La elaboración de una norma de control
- El establecimiento de diversos procedimientos para supervisar
- Capacitación y divulgación
- Desarrollo de un programa de investigación y de estudios complementarios a este trabajo que resultaron, algunos de ellos, imprescindibles.
- La implementación efectiva del programa de pretratamiento de descargas
- Control de la erosión en suelos de conservación principalmente
- El cumplimiento de la NOM 004 ECOL-2001 en las PTAR's

Además, en este trabajo se presenta, ya sea en el cuerpo del mismo o en anexos, las técnicas analíticas empleadas, los procesos de tratamiento aplicables a los azolves, un manual de operación adecuado para la SMADF, el contenido técnico para la redacción de la norma requerida, el diseño de un foro de consulta y los mecanismos de financiamiento para los trabajos por realizar, entre otras cosas.

## 2.2 DEFINICIÓN DEL PROBLEMA

### 2.2.1 IMPORTANCIA SOCIAL

Con el objeto de evitar inundaciones y mantener la capacidad del sistema hidráulico de la Ciudad de México, la Dirección General de Construcción y Operación Hidráulica

(DGCOH) realiza periódicamente el desazolve de presas, lagunas de regulación, cauces y líneas de conducción de aguas residuales producidas en el Distrito Federal. Además, opera 21 de las 27 plantas de tratamiento de aguas residuales de la Ciudad de México que maneja el gobierno. De éstas es responsable también del tratamiento, manejo y destino final de los lodos producidos.

Dada la elevada demanda de terreno que implica depositar los azolves en el relleno sanitario, en un futuro se dificultará la disposición en dicho sitio cuya capacidad se debe preservar prioritariamente para los desechos sólidos, ya que se estima que tiene una vida útil hasta febrero de 2004. El Bordo Poniente es el relleno más grande de América Latina, recibe 12,500 ton/día, lo que por una compactación de 1000 $Kg/m^3$ (GDF-DGSU, 2002), implica que los 656 000 $m^3$/año de azolves producidos representa el 14% de su capacidad anual del relleno.

La Secretaría del Medio Ambiente del Distrito Federal (SMA-DF) es quien tiene a su cargo la autorización del vertido y/o aprovechamiento de estos materiales, por lo que es necesario conocer adecuadamente el problema de disposición o los usos posibles de los materiales.

La evaluación de diversos parámetros físicos, químicos y microbiológicos servirá para clasificar los materiales y establecer criterios de manejo y destino final. Aún cuando varios de los parámetros analizados no se encuentran normados, algunos metales y los parámetros microbiológicos cuentan con límites en el Proy-NOM-004-ECOL-2001 publicado por la Secretaría del Medio Ambiente y Recursos Naturales (SEMARNAT).

### 2.2.2 REQUERIMIENTOS TÉCNICOS

En el convenio que regula el proyecto "PROGRAMA DE MANEJO AMBIENTAL DE LOS RESIDUOS PROVENIENTES DE DESAZOLVE DEL SISTEMA DE DRENAJE DEL DF Y DE LAS PRESAS, ASÍ COMO DE PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES MUNICIPALES", presenta todos los requerimientos solicitados y acordados por ambas partes. Con relación al desarrollo de un sistema informático, que es el tema único de esta tesis, la Secretaría del medio Ambiente indica en el "ANEXO ÚNICO TÉRMINOS DE REFERENCIA", los alcances de este sistema, los cuales se presentan a continuación bajo el titulo de "Desarrollo de un banco de información":

### 2.2.3 DESARROLLO DE UN BANCO DE INFORMACIÓN

Se elaborará un banco de información y difusión, que deberá contener por lo menos lo siguiente:

- Introducción, antecedentes, descripción del programa, actividades del programa, alcances, beneficios ambientales y de salud, datos técnicos de generación, tipo de lodos y azolves y su composición, destinos y usos potenciales, definiciones, participantes en el programa, gráficos, tablas, fotos.
- La base de datos debe ser diseñada en un *DBMS* comercial, y formatos de consulta para páginas Web.

- Capacitación de dos personas para el manejo de la base de datos.
- La información que contendrá la base de datos, será seleccionada y aprobada por la Secretaría del Medio Ambiente con la participación del grupo de trabajo.
- El nivel de acceso al contenido del banco de información por instancias ajenas al grupo de trabajo conformado para los fines de este instrumento, será definido por la Secretaría del Medio Ambiente.

En base a estos requerimientos se plantean los componentes básicos que se presentan en la Tabla 2-3, Estos puntos han sido estudiados y aceptados por la Secretaría del Medio Ambiente.

**Tabla 2-3 Secciones que formarán parte de la solución Web propuesta.**

| Sección | Descripción | Estructura |
|---|---|---|
| Introducción | Breve descripción del objetivo del proyecto y comentarios diversos de los encargados del proyecto | Solo texto |
| Manejo de la información | Parte central de la solución propuesta. | |
| | Acceso a la información por 3 métodos: | Mapa de navegación.<br>Listado de los sitios en forma de texto.<br>Búsqueda directa en la base de datos. |
| | Despliegue, manipulación y generación de reportes, a partir de la información solicitada. | Despliegue de la información en texto sin formato o en reportes. |
| | Posibilidad de acceder a datos históricos de los parámetros de evaluación de azolves. | |
| | Presentará el procesamiento de la información basado en el algoritmo diseñado por el M. en I. Juan Manuel Méndez C. (IINGEN-UNAM 2002). | Procesamiento del lado del servidor. |
| | Acceso a las Normas Ambientales Usadas en el proyecto. | |
| Acervo de Informes | Presentación de resultados, gráficas, presentación e informe final. | Documentos en formato comprimido. |
| Acervo de Imágenes y video | Presentación del acervo fotográfico y multimedia. | Documentos en formato JPG y WMV |
| Ayuda e información | Créditos, manuales, soporte y referencias relacionados con el tema. | Solo texto y documentos en formato comprimido |
| Recomendaciones | Presentación de posibles soluciones a los problemas que presente cada sitio de interés | Solo texto y documentos en formato comprimido |
| Administración | Manejo de cuentas de usuarios<br><br>Mantenimiento y soporte a la base de datos | Acciones controladas por medio de procedimientos almacenados y formularios Web. |

En el caso de nuestro cliente, el valor agregado que buscan ofrecer a sus clientes, reside en la alta disponibilidad de los datos e información que esta Secretaría recaba de sus diversos departamentos. Un ejemplo de estos esfuerzos se ve materializado en el programa: "RED AUTOMÁTICA DE MONITOREO ATMOSFÉRICO DE LA ZONA METROPOLITANA DE LA CIUDAD DE MÉXICO", el cual tiene como finalidad presentar la información de los contaminantes y parámetros meteorológicos medidos por la Red Automática de Monitoreo Atmosférico. La información se almacena por estación de monitoreo en unidades de medición. Únicamente, los contaminantes que tienen Normatividad Oficial Mexicana (NOM) de efectos a la salud son transformados al Índice Metropolitano de la Calidad del Aire (IMECA); por lo que, el IMECA se basa en dicha normatividad para su cálculo. Su consulta se puede realizar en la siguiente dirección electrónica " http://148.243.232.103/imecaweb/ ".

Aplicaciones con esta naturaleza, marcan la filosofía de nuestro cliente. Para que soluciones como la antes mencionada se encuentren disponibles el mayor tiempo posible, es necesario un equipo técnico y humano mínimo para solventar las necesidades del medio ambiente de la aplicación, por tal motivo es deseable contar con un estudio previo sobre la infraestructura, recursos humanos y materiales con los que se dispondrán.

La necesidad que expresamos de contar con el estudio antes mencionado es debido a que partiendo de él se podrán realizar los ajustes necesarios, tanto a la infraestructura como a la aplicación misma o al equipo humano que será responsable de su conservación. A continuación plantearemos los recursos necesarios para nuestra aplicación.

## 2.3 INFRAESTRUCTURA NECESARIA

### 2.3.1 ELECCIÓN DE ARQUITECTURA

Como se puede ver en el capítulo 1 la arquitectura más adecuada para el desarrollo de nuestra solución es una basada en la arquitectura cliente/servidor debido a su flexibilidad, interoperabilidad y escalabilidad, características deseables en nuestro sistema. Dentro de esta arquitectura vimos que el diseño más adecuado a nuestras necesidades es la arquitectura de n-capas basada en clientes delgados lo cual nos permitirá tener una diferenciación más clara entre las distintas capas de nuestra aplicación. Este tipo de arquitectura es la más empleada en todo el mundo hoy en día especialmente en las aplicaciones basadas en Web, el cual será nuestro caso. Con este tipo de arquitectura la capa de presentación será colocada en la PC cliente las reglas de negocios en nuestro servidor central, una capa de acceso a los datos que se encargue de las consultas y la capa de datos (base de datos) en otro servidor. Esta división nos permitirá modificar código en caso necesario en un sólo lugar de forma que se mejoren las labores de mantenimiento. Aún cuando estas capas estén separadas, cada una depende de las otras dos para que la aplicación actúe como una sola entidad. El hecho de que la capa de presentación se ejecute sólo en la PC cliente, permite que los recursos del cliente no sean muy avanzados, esto es particularmente necesario en nuestro caso, debido a que se desconoce el número de usuarios que accederán al sistema y las características del equipo cliente, por lo que será

necesario aumentar la carga de trabajo en los servidores para no tener que preocuparnos demasiado por el desempeño en el cliente que consuma nuestra aplicación.

Siguiendo este diseño y en caso de que en un futuro se planteara el escalamiento de nuestro sistema, solo debemos hacer unas pequeñas modificaciones en algún módulo específico teniendo un impacto mínimo en las demás capas, logrando de esta forma un mayor control de cada componente y la posibilidad de reutilizar código, además de brindarnos más control sobre la seguridad en el acceso a los datos al implementar diversos niveles de autenticación en distintas capas, con lo que se asegura la integridad de los datos.

Una aplicación basada en Web requiere al menos tres recursos básicos: un equipo servidor, una red de comunicaciones y un grupo de soporte. Existen más recursos que podemos listar, pero debido a la naturaleza del proyecto, nosotros consideramos que los elementos antes mencionados son los básicos para garantizar el entorno de nuestra aplicación. Las características de los recursos empleados durante la fase de desarrollo y los propuestos para garantizar el correcto funcionamiento de la aplicación que estamos desarrollando se presentan a continuación.

### 2.3.2 SERVIDOR

Cualquier aplicación basada en Web implica revisar los recursos disponibles en el servidor Web de producción, durante el desarrollo de la aplicación es común trabajar en uno o varios servidores de pruebas, los cuales suelen estar dedicados a un par de proyectos. La anterior práctica conlleva a los grupos de desarrollo a pensar que el servidor de producción ofrecerá el mismo desempeño que el servidor de desarrollo. Pensamiento que conduce a modificaciones y reformas al producto final para ajustarse a los recursos con los que se cuentan.

Al no contar con la información referente a los recursos existentes en el o los servidores de producción de nuestro cliente, se ha pensado en diseñar una aplicación ligera, que sea soportada por equipos de cómputo básicos en comparación con un servidor de hardware especializado.

Las siguientes tablas presentan las características de hardware para el desarrollo, implementación y consulta de la solución Web. Presentamos las características mínimas para su funcionamiento y además características que recomendamos para un mejor desempeño.

#### 2.3.2.1 EQUIPO DE DESARROLLO

Será el equipo donde se lleve a cabo la formulación de la solución Web, desarrollo, pruebas, análisis, trabajo en equipo del desarrollador, analista, y demás personal que se involucre en el producto final. El número de equipos para desarrollar son 2, lo que permite tener un tiempo de desarrollo más corto y facilita el trabajo de los integrantes, además de facilitar un ambiente de pruebas de acceso a la solución, manejo de administración, servidor Web, servidor de base de datos, cliente y diseño en paralelo. El desarrollo de la

aplicación se ve beneficiada en tiempo y costo con un número mayor de equipos y desarrolladores, por lo que se recomienda más personal y equipo.

La columna de requerimientos del sistema está basada en características suficientes para el desempeño de tareas de edición, codificación, soporte de servicios, sistema de pruebas, emulación de cliente y servidor. Permitiendo al equipo de desarrollo realizar la tarea de formulación de la solución Web.

**Tabla 2-44 Características de Hardware del equipo de desarrollo**

| Componente | Requerimientos del Sistema | Equipo Recomendado |
|---|---|---|
| Procesador | Pentium III a 500 MHz | Pentium IV a 1 GHz o superior |
| Memoria | 256 MB | 512 MB o mas |
| Disco Duro | 20 GB | 20 GB |
| Sistema Operativo | Windows 2000 Profesional | Windows 2000 Profesional |
| Software | Suite de desarrollo Visual Studio .Net Professional | Suite de desarrollo Visual Studio .Net Professional |
| Periféricos | Tarjeta de Red de velocidad 10/100 Mbps | Tarjeta de Red de velocidad 10/100 Mbps |
| Video | Súper VGA con resolución 800 x 600 píxeles | Súper VGA con resolución superior a 800 x 600 píxeles |
| Servicio | Servicio de conexión a Internet | Servicio de conexión a Internet |

Las características recomendadas, son sugeridas para poder tener un mejor desempeño en tiempo y costo de desarrollo, debido a tiempos de edición y simulación de pruebas que requieren de un mejor desempeño por parte del equipo, además de agilizar el trabajo en equipo y de tareas en conjunto; ya que el proceso de edición final presenta pruebas de carácter gráfico y es necesaria la intervención de procesos de edición gráfica, código, y simulación del cliente y servidor.

La plataforma de desarrollo Windows es seleccionada debido a la alta integridad de las herramientas de desarrollo, sistema de almacenamiento y manejo de datos, ya que se presenta un mejor desempeño que en otras plataformas.

La elección de la *Suite* de desarrollo *Visual Studio .Net Professional* se encuentra respaldada debido a una mejor integración de las herramientas de desarrollo con la plataforma Web (*IIS*), con los servicios Web, una mejor integración a la plataforma de *SQL* Server, herramientas de desarrollo estructurado y seguridad en la fase de desarrollo. Esta plataforma proporciona herramientas con las siguientes características:

- Alta integración con sistemas de Bases de datos
- Rápido desempeño de soluciones Web
- Codificación y sistemas de apoyo que facilitan la programación
- Soporte para varios lenguajes de programación
- Soporte para migrar y dar apoyo a otras plataformas en aspectos de Sitios Web
- Curva de aprendizaje baja
- Formulación de la solución en documentos *ASPX*

*ASPX* es el formato de archivos para contener formularios de datos para aplicaciones Web, debido a la integración del servicio Web y el *Framework* permiten la interacción del

usuario con la aplicación de forma optimizada, simple y fácil de emplear debido a un ambiente gráfico.

El empleo de conexión a red por medio de un *modem* o tarjeta de red, es para uso de pruebas de clientes remotos, prueba de servicios, conexión al servidor de producción, para manejo de datos y servicios y búsqueda de información. Esta conexión es dependiente del ISP.

### 2.3.2.2 EQUIPO DE PRODUCCIÓN

El equipo de producción es el que proporcionará los servicios de Web, Base de Datos, y componentes adicionales para el funcionamiento de la solución Web, este equipo dedicado requiere de características especiales de un servidor, por lo que las listamos en la Tabla 2-55.

**Tabla 2-55 Características de Hardware equipo de producción**

| Componente | Requerimientos del Sistema | Equipo Recomendado |
|---|---|---|
| Procesador | Pentium IV a 2.4 GHz | Procesamiento dual Xeon a 2.0 GHz |
| Memoria | 512 MB | Superior a 512 MB |
| Disco Duro | HD SCSI con capacidad de 40 GB | 2 HD SCII con capacidad de 40 GB en configuración RAID 1 |
| Sistema Operativo | Windows 2000 Server | Windows 2000 Server |
| Software | IIS 5 o versión posterior<br>FrameWork 1.1 | IIS 5 o versión posterior<br>FrameWork 1.1 |
| Periféricos | Tarjeta de Red 10/100 Mbps | Tarjeta de Red 10/100/1000 Mbps |
| Video | Súper VGA con resolución 800 x 600 píxeles | Súper VGA con resolución 800 x 600 píxeles |
| Servicio | Servicio de conexión a Internet de 128 Kbps | Servicio de conexión a Internet, con velocidad de 256 Kbps o superior |

El empleo de un equipo dedicado al Web y al alojamiento de la base de datos es recomendado, con el fin de obtener un mejor desempeño y rendimiento para el proceso de acceso por medio Web y en la comunicación de la aplicación con el servidor de base de datos para el manejo de datos.

Los componentes listados en la Tabla 2-55 de Requerimientos del sistema permiten tener un desempeño de la aplicación de forma suficiente para un número de 100 usuarios de una forma satisfactoria, pero para tener un mejor desempeño y lograr tener un aplicación de alta disponibilidad a un número mayor de usuarios y de forma concurrente se recomienda tener un equipo con las características recomendadas.

El empleo de *IIS* 5 y del *Framework*, permite tener un mejor desempeño en la plataforma *Windows* y una mejor integración con los servicios de comunicación con la base de datos *SQL Server*, ya que cuenta con un mayor desempeño en velocidad, seguridad nativa, un medio de acceso controlado en el que sólo la aplicación Web puede tener acceso y que proporciona seguridad, limitando el acceso a personas no autorizadas. Este esquema de trabajo permite no sólo al desarrollador simplificar las tareas de control y enlace de la solución Web con la base de datos, sino que permite una mejor administración de la aplicación así como su mantenimiento, actualización y auditoria en cuanto al flujo de datos internos.

El empleo de otro servidor Web no permite el manejo de forma transparente y confiable de la aplicación así como de los usuarios, además de requerir un mayor tiempo de compilación de los documentos *XML* y mostrar falta de soporte para documentos *ASPX*.

### 2.3.2.3 EQUIPO CLIENTE

Este equipo es el que se empleará para poder consumir la aplicación, este equipo para uso de los usuarios finales no requiere de características especiales ya que hoy en día son muy fáciles de encontrar o que en gran parte del mercado son accesibles.

**Tabla 2-66 Características de Hardware del equipo cliente**

| Componente | Requerimientos del Sistema | Equipo Recomendado |
|---|---|---|
| Procesador | Pentium II a 300MHz, similar o superior | Pentium III a 800 MHz, similar o superior |
| Memoria | 64 MB | 128 MB o mas |
| Disco Duro | La aplicación no requiere alojar datos en el equipo cliente | La aplicación no requiere alojar datos en el equipo cliente |
| Sistema Operativo | Cualquier plataforma | Cualquier plataforma |
| Software | Navegador Web, Explorer 5 o Netscape 5 | Navegador Web, Explorer 6 o Netscape 5 o superior |
| Periféricos | Modem, Tarjeta de red 10/100 | Modem, Tarjeta de red 10/100 |
| Video | Súper VGA con resolución 800 x 600 píxeles | Súper VGA con resolución 800 x 600 píxeles o más |
| Servicio de Internet | Cualquier servicio de conexión a Internet | Cualquier servicio de conexión a Internet |

La arquitectura del procesador y tecnología recomendada son las mostradas en la Tabla 2-66 y permiten tener por parte del usuario final un buen desempeño en cuanto al uso de la solución Web, ya que sólo realiza acciones de peticiones de red, despliegue de gráficos y texto, esto se ve beneficiado de acuerdo al desempeño del equipo y de las demás aplicaciones que en ese momento se encuentren en uso, siendo este último punto, algo muy cotidiano que se realiza en procesos de investigación y consulta en Internet. Los recursos del equipo mencionado sólo son para tener un buen desempeño en la consulta y acceso a la aplicación, ya que ésta no realiza un consumo de memoria, espacio en disco duro para poder realizar operaciones o consultas para el funcionamiento del Servicio Web.

La plataforma del sistema operativo del cliente no es crucial para el uso de la aplicación, debido a que sólo es requerido un navegador Web actualizado que cuente con el soporte para páginas *ASPX*, por lo tanto puede ser empleado cualquier sistema operativo con soporte visual y navegador Web.

El servicio de conexión a Internet es necesario para poder tener acceso a la aplicación debido a que esta se encuentra alojada como un Servicio Web, siendo el caso para usuarios finales que requieran tener acceso y se encuentran fuera de la institución o empresa. Para el caso de una Intranet, los usuarios finales pueden tener el acceso por medio de la misma Intranet sin necesidad de acceso a una red externa, debido a que se puede tener acceso solo al Web interno que señale a la aplicación.

### 2.3.3 RED DE COMUNICACIONES

La red de comunicaciones desde hace unos años es un recurso que día con día se convierte en elemento crítico para las operaciones de las empresas. Este recurso permite la cómoda distribución de información y datos diversos entre las empresas y los clientes o entre los clientes y los socios o filiales de la misma.

La implementación de una aplicación basada en Web, puede llegar a consumir todos los recursos disponibles en la red de comunicaciones una vez que ésta se encuentre en producción. Este problema puede llegar a replantear totalmente los servicios y recursos ofrecidos por la aplicación y en casos más drásticos implica la posibilidad de replantear la infraestructura completa de comunicaciones, ambas acciones implican tiempo e incremento en los presupuestos otorgados.

Es deber del líder de proyecto solicitar la información referente a la tecnología, ancho de banda contratado, uso del ancho de banda y futuro de los recursos de comunicación disponibles. El desarrollo de la aplicación se llevó a cabo en el entorno de red del Instituto de Ingeniería de la UNAM donde se cuenta con una estructura con las siguientes características.

- Conexión T1 a la Red UNAM
- Velocidad en red interna 10 Mbps

Una red con estas características permite la implementación del sistema, pruebas y desarrollo de la aplicación en un ambiente estable, facilitando la consulta de la misma desde distintos lugares y subredes distintas, con fines de prueba de comunicación, desempeño y funcionalidad.

El ambiente del entorno de red y características del Gobierno de la Ciudad de México en cuanto a su estructura, no fue proporcionado, hecho que nos llevo a plantearnos algunos casos donde los recursos de la red de comunicaciones permitan, alcanzar un desempeño razonable para el acceso a la aplicación Web.

**Tabla 2-77 Casos de Servicio de conexión**

| | Caso 1 | Caso 2 | Caso 3 |
|---|---|---|---|
| Tipo de Conexión | ADSL | ADSL | ADSL |
| Velocidad | 128 Mbps | 256 Mbps | 512 Mpbs |
| Tipo de Dirección IP | Homologada | Homologada | Homologada |

El desempeño de cada una de estos casos se ve afectado por el flujo de datos que se encuentre en ese momento en el canal de comunicación, pero su rendimiento no se ve gravemente afectado, estos casos demandan el empleo de una dirección de tipo homologada para efectos de ser consultada desde cualquier punto geográfico, siempre y cuando se cuente con conexión a Internet.

Para efectos de consulta dentro de una Intranet se puede tener una configuración por parte del servidor Web con una dirección no homologada, considerando que esta sea alcanzable por los equipos cliente.

### 2.3.3.1 PERSONAL DE SOPORTE Y ADMINISTRACIÓN

Toda empresa que cuente con un servicio autónomo de publicación en Internet ("Alojamiento de un sitio Web"), dispone de los servidores y del personal capacitado para la administración de la infraestructura presente. El perfil del personal dedicado a la administración de los servidores por lo regular solo abarca el conocimiento de la operación y seguridad de los servidores internos ("Servidores de dominio y operaciones internas de la empresa"), los servidores de bases de datos y los servidores Web.

El equipo de administradores permitirá que nuestra aplicación cuente con un servidor o conjunto de ellos permanente, en "línea". En el caso de que el cliente especifique que se cuenta o se desea conformar un grupo de soporte al código de nuestra aplicación, sus labores abarcarán la realización de modificaciones y correcciones al código, así como escalar la aplicación. Este aspecto requiere de gente con conocimientos en la tecnología usada, así como completo entendimiento de la filosofía usada en la construcción de esta solución.

### 2.3.3.2 ALOJAMIENTO EXTERNO

Durante toda la etapa de desarrollo, la aplicación fue probada en un medio ambiente totalmente distinto al que ofrecerá nuestro cliente, razón por la cual el rendimiento final de la aplicación representa una incógnita, los puntos antes señalados nos permitirán diseñar una aplicación que se ajuste lo más posible a un ambiente con iguales o menores recursos. Una etapa de desarrollo con estas características deja abierta la posibilidad de pensar en la renta de un alojamiento externo, servicio que para muchos clientes representa una posibilidad de recortar gastos en expansión de servidores o incrementos en sus costos de comunicaciones y personal de soporte. Desde nuestro punto de vista, la renta de un alojamiento para una aplicación basada en Web, es una muy buena opción para las empresas pequeñas que no pueden realizar inversiones considerables en sus áreas críticas; antes de optar por un alojamiento externo, se deberá de evaluar ¿Qué tan crítica es la

información y datos que serán manejados por la aplicación? En la medida que la aplicación maneje información crítica de la empresa o sus clientes, la opción de un alojamiento externo se debe descartar y evaluar medios de financiamiento para solventar los gastos de expansión de las áreas críticas.

#### 2.3.3.3 DESEMPEÑO DE LA APLICACIÓN

La aplicación se basa en la recopilación de los datos proporcionados por distintos procesos, análisis y pruebas de laboratorio de cada muestra de estudio, lo que implica que estos datos son de carácter lógico, numérico, o de valores subjetivos. Además que estos no son recopilados en un mismo paso, debido a la naturaleza y costo de estas pruebas se lleva a cabo en varias horas y pasos de acuerdo a la metodología de estudio de aguas residuales.

La aplicación requerirá un alojamiento centralizado con el fin de proporcionar seguridad, disponibilidad y control de los datos del proyecto, por lo que se establece la necesidad de un equipo y un sistema centralizado con el alojamiento de los datos.

El acceso de los datos por parte del usuario requiere de una disponibilidad de los datos sin importar el horario o la situación geográfica debido a la necesidad de tener un centro de recopilación accesible desde alguna localidad que necesite tener acceso a los datos para consulta o modificación de los mismos.

La información almacenada por el sistema debe contar con un sistema de seguridad que garantice la integridad de la misma aplicación, los datos, y el acceso, debido a que la naturaleza del estudio de aguas residuales demanda una alta discreción y confidencialidad de los datos y del proyecto.

### 2.3.4 ANÁLISIS DE DATOS

Con base a los datos recopilados se tiene el análisis de los siguientes puntos, para el diseño de la aplicación en general.

#### 2.3.4.1 CONCENTRACIÓN DE DATOS

La recopilación de datos y concentración, demanda el empleo del esquema Cliente/Servidor, para el uso de un servidor dedicado para alojar datos bajo un sistema de seguridad y disponibilidad de la aplicación. Y de un cliente capaz de poder tener acceso al sistema de forma segura y rápida para poder realizar cambios, consultas, y manejo de cuentas de usuario, dando como resultado los siguientes parámetros:

- Servidor dedicado para el alojamiento de datos.
- Servicio de acceso, control, seguridad de datos.
- Sistema de respaldo y manejo de datos.
- Integrable con sistemas existentes y medios de almacenamiento básicos para efectos de reporte, y análisis en escritorio.

### 2.3.4.2 DISPONIBILIDAD

La disponibilidad de la aplicación requiere que los usuarios puedan tener acceso a los datos sin la necesidad de tener un cliente estacionario que dependa de la estructura de red corporativa o de la dependencia, debido a que los sitios de muestreo, laboratorios, así como el centro de almacenamiento se encuentran localizados en lugares distintos. Esto implica la necesidad de equipo necesario para la aplicación, alojamiento de datos, redundancia, acceso por parte de los clientes de forma rápida, confiable, y segura, sin la necesidad de una configuración en especial para poder tener acceso a la aplicación. Resumiendo los siguientes parámetros:

- Disponibilidad de la aplicación funcional.
- Disponibilidad por parte de los equipos clientes sin necesidad de una conexión dedicada.
- Disponibilidad desde otra localidad fuera del centro de recopilación de datos.
- Disponibilidad a partir de equipos cliente con pocos recursos.
- Interfaz para el enlace entre el Servidor y Cliente

### 2.3.4.3 SEGURIDAD

La seguridad es requerida tanto en el servidor, cliente y el transporte de los datos entre estos dos elementos, así como en la aplicación misma. Dado esto, se tienen las siguientes consideraciones necesarias para la seguridad del sistema:

- Cuentas de usuarios y administración.
- Contraseña.
- Seguridad en la base de datos.
- Seguridad de acceso a la aplicación.
- Control de acceso a datos.
- Sistema de almacenamiento de respaldo y restauración.

Los medios de seguridad serán dependientes de cada módulo implicado en la solución del problema, donde el sistema de comunicación contendrá su propio medio de seguridad, la base de datos contendrá su sistema de seguridad, y el medio de enlace y validación también contendrá su sistema de seguridad.

## 2.3.5 MODELO DEL SISTEMA

¿Por qué se realiza un modelo de la aplicación?

El modelo del Sistema a desarrollar se basará en la implementación de Técnicas de desarrollo de software bajo el Lenguaje Unificado de Modelado (UML), el cual proporciona las herramientas necesarias para el desarrollo de una aplicación de Software dentro del ambiente de red, así como proporcionar facilidad en su seguimiento y documentación, permitiendo dar al cliente las herramientas necesarias para futuras modificaciones o actualizaciones del sistema en caso de ser necesario.

### 2.3.6 PROCESO UNIFICADO

El desarrollo de un sistema de software requiere hoy la necesidad del control y manejo de cada una de las etapas desarrolladas bajo un grupo de desarrolladores o personal de apoyo, ya que la importancia y calidad del software es medido de acuerdo a la metodología empleada para el desarrollo. Es por eso la necesidad de cubrir los siguientes puntos:

- Poseer una guía para ordenar las actividades de un equipo de desarrollo.
- Dirigir las tareas de cada desarrollador por separado y del equipo como un todo.
- Especificar los módulos que deben desarrollarse.
- Ofrecer criterios para el control y medición de los productos y actividades del proyecto.

Donde podemos tomar la definición del Proceso Unificado como: "Proceso de desarrollo de software basado en el Lenguaje Unificado de Modelado, y que es iterativo, centrado en la arquitectura y dirigido por los casos de uso y los riesgos. Proceso que se organiza en cuatro fases: Inicio, Elaboración, Construcción y Transición, y que se encuentra en torno a cinco flujos de trabajo fundamentales: recopilación de requisitos, análisis, diseño, implementación y pruebas. Proceso que describe en términos de un modelo de negocio, el cual esta a su vez estructurado en función de tres bloques de construcciones primordiales: trabajadores, actividades y artefactos."[1]

El Proceso Unificado hace uso del Lenguaje Unificado de Modelado (UML) para poder preparar los esquemas de un sistema de Software interconectados a través de interfaces.

De aquí la importancia de definir el Lenguaje Unificado de Modelado, ya que será con él cómo se llevara a cabo la elaboración, el análisis, la puesta en marcha de la aplicación bajo un control del Proceso Unificado tomando en cuenta los recursos con los que se cuenta y lo necesario para su ejecución.

#### 2.3.6.1 UML

"El Lenguaje Unificado de Modelado (Unified Modeling Language, UML) es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software"[2]

UML es un sistema adecuado para el desarrollo modelos de las aplicaciones o soluciones de información e incluso aplicaciones distribuidas como son las basadas en Web, debido a que

---

[1] Ivar Jacobson, et al, Op. Cit. p.431

[2] El Lenguaje Unificado de Modelado, Grady Booch, James Rumbaugh, Ivar Jacobson, Edit. Addison Wesley, p11

permite expresar la aplicación en cada uno de sus componentes, es simple de implementar debido al sentido de representación gráfica que permite al usuario entender la aplicación a desarrollar, así como permitir mostrar componentes que inicialmente no se encuentran contemplados.

El desarrollo bajo UML involucra el manejo de cada uno de los componentes así como las tareas o procesos que se desarrollan en cada una de las etapas del producto, tarea que continuamente se presenta en el desarrollo de una aplicación, pero bajo el apoyo de UML se tienen aspectos básicos que permiten tener una visión general del proyecto.

UML es un lenguaje para:

- Visualizar
- Especificar
- Construir
- Documentar

Los artefactos, clases y objetos que conforman cualquier aplicación en desarrollo.

Se puede decir que UML es un lenguaje ya que posee una estructura formal que permite llevar a cabo una integración de varios tipos de análisis con el fin de ser entendibles de forma natural al emplear esquemas y relaciones simples de comunicación entre cada uno de los componentes o artefactos y sus relaciones lógicas que permiten el funcionamiento del sistema.

El empleo de UML hace uso de técnicas ya conocidas dentro del desarrollo de Software o de la Ingeniería de Software, ya que toma los principales elementos del desarrollo de una aplicación de computación, dentro de un ciclo de vida clásico y lo complementa con una especie de ciclos o iteraciones dentro de cada una de sus etapas, con el fin de simplificar las tareas de los diseñadores y demás personal implicado en el desarrollo de la aplicación. Teniendo un control de cada uno de los bloques desarrollados UML nos muestra un desarrollo general como ilustra la Figura 2-1

Se pueden observar cuatro fases fundamentales: Inicio, Elaboración, Construcción y Transición. Las cuales presentan típicamente iteraciones en cuanto a su ejecución debido a las modificaciones, cambios que son requeridos por el sistema, usuario o por los requisitos del sistema. Las tareas típicas por parte del equipo de desarrollo serían las siguientes: Recopilación de Requisitos, Análisis de los datos y del problema, Diseño de la solución, Implementación y Prueba de los componentes que conforman la aplicación

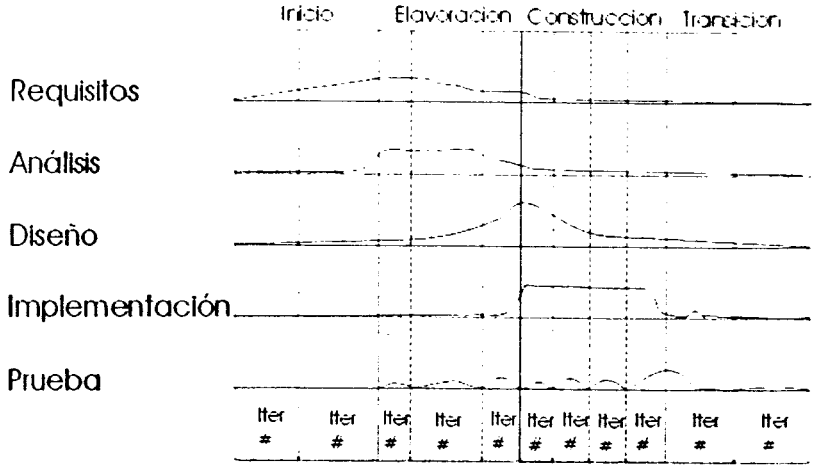



**Figura 2-1 Diagrama de procesos de desarrollo**

Justificación del empleo de UML y el Proceso unificado

UML nos permite desarrollar cualquier tipo de aplicación en un entorno de pequeñas y grandes aplicaciones, proporcionando las herramientas necesarias para su desarrollo a lo largo de su Vida de Software, permitiendo:

- Integración de aplicación
- Documentar el desarrollo de la aplicación
- Control de la arquitectura empleada
- Sistematización del desarrollo
- Facilidad de implementación
- Visualización de la aplicación en su entorno
- Especificación de forma detallada de los componentes que conforman la aplicación
- Apoyo a la construcción de aplicaciones y componentes necesarios para la solución o proyecto
- Adecuación para sistemas Web, debido a su integridad con sistemas distribuidos
- UML no es un lenguaje visual, pero sus modelos se pueden integrar a varios sistemas o lenguajes de programación

El empleo de UML permite a cualquier persona que se encuentre dentro del desarrollo del sistema poder consultar y comprender los componentes que conforman la aplicación, permitiendo una mejor integración de las tareas a desarrollar por los desarrolladores, analistas, arquitectos, ingenieros de pruebas, clientes, usuarios y demás participantes. UML es un lenguaje de desarrollo no enfocado a un sistema o lenguaje de programación, sino un lenguaje de integración de técnicas de desarrollo, análisis, búsqueda, reingeniería, ingeniería e implementación, basado en aspectos esquemáticos que facilitan la comprensión de cada uno de los componentes de cualquier sistema.

La implantación de UML para el desarrollo de la aplicación se funda en la integridad de los sistemas participantes, es decir, debido a que la propuesta de una solución Web que

requiere composición con los servicios Web, comunicación y una arquitectura de Cliente/Servidor es necesario un análisis de cada uno de ellos y sus estrechas relaciones para el mejor desempeño de la solución propuesta.

El empleo de otra técnica de Ingeniería de software como los tradicionales ciclos de vida Clásica o prototipos, requieren distintas características para poder ser implementados, siendo las siguientes algunas de sus características y desventajas:

- Ciclo de Vida clásica. El empleo de una continuidad lineal del desarrollo de la aplicación, no permite el desarrollo de tareas en forma paralela para el grupo de desarrolladores y analistas, ya que se requiere de un análisis por cada componente de la aplicación, invirtiendo tiempo, y gastos de estudio.
- Prototipos. Un sistema de prototipos proporciona soluciones en tiempos cortos, pero requiere de una constante participación con el cliente debido a las constantes consultas del desempeño de la aplicación así como el control de la aplicación un poco fuera del formalismo de desarrollo, implicando falta de control, calidad y problemas que pueden surgir después de la entrega del producto debido a la falta de integración con otros sistemas y modificaciones futuras.
- El empleo de UML en resumen nos brinda control del desarrollo, simplicidad de implementación, cubrir con las necesidades del cliente, proporciona una arquitectura flexible para la integración con otros sistemas existentes y nuevas actualizaciones, así como el ahorro en el desarrollo del sistema en cuanto a análisis, personal, y gastos de implementación de la infraestructura requeridas para el desarrollo de la Solución Web.

## 2.4 ARQUITECTURA DE LA APLICACIÓN A DESARROLLAR

La arquitectura propuesta bajo UML nos permite obtener una perspectiva dentro del ciclo de vida de Software, un control en su desarrollo, recopilación de información, análisis de funcionalidad, documentación de la aplicación, planeación y justificación de los elementos que interfieren para la generación de la aplicación.

- Vista de Caso de Uso
- Vista de Diseño
- Vista de Implementación
- Vista de Procesos
- Vista de Despliegue

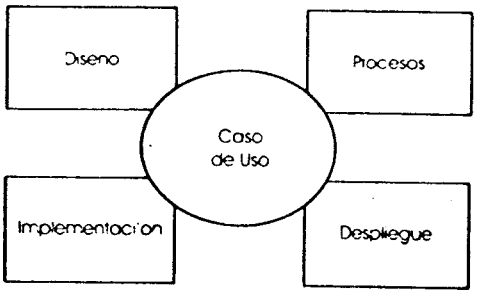



**Figura 2-2 Arquitectura a Implementar**

A continuación se desarrollarán los distintos puntos de vista para el desarrollo de la aplicación, para cada uno de los puntos que son requeridos para el punto de vista de procesos, diseño, despliegue y casos de uso.

### 2.4.1 MODELADO DE CASO DE USO

El modelado de la aplicación Web emplea los siguientes componentes, los cuales son casos necesarios y requeridos por el sistema para la ejecución del sistema solicitado por el proyecto de Aguas Residuales:

- **Usuario**, es una entidad que interactúa con el sistema, el cual proporciona entradas al sistema el cual le proporciona resultados de acuerdo a las secuencias lógicas programadas por medio de una interfaz o medio de salida de los datos.
- **Validar**, para poder ingresar al sistema es necesario la validación del usuario, así como para el control de los datos, la aplicación y seguridad.
- **Consultar Datos**, esta actividad permite la consulta de los datos del proyecto de alguna instancia en particular: presa, valores de calidad del agua, ríos, o datos correspondientes a la aplicación como sus normas y funcionalidad
- **Manejo de Cuentas**, la administración de la solución Web requiere de la administración de los usuarios, por lo que se requiere un sistema de administración de las cuentas que pueden hacer uso de la aplicación.
- **Manejo de Datos**, la manipulación de los datos requiere de las actividades básicas de acceso, actualización, ingreso o eliminación de datos, es debido a eso la necesidad de un módulo de manejo de datos.
- **Obtener Datos**, las consultas realizadas por los usuarios, demanda adquirir los datos para efectos de reporte y de consulta posterior, por lo que se presenta la necesidad de obtener los datos en un archivo con formato.
- **Consulta de Ayuda**, la aplicación proporciona una ayuda básica para su uso.
- **Impresión**, esta actividad es requerida para proporcionar un reporte impreso de los datos, cuentas o reportes seleccionados por el usuario.
- **Mantenimiento del sistema**, esta tarea es exclusiva de la parte administrativa, donde se puede tener un manejo básico del estado de la aplicación y de los datos.

### 2.4.1.1 LIMITANTES

La aplicación Web, proporciona una serie de limitantes en su desempeño, esto debido a la naturaleza de las tareas mismas o su uso en específico, por lo que a continuación se citan las limitantes de cada caso de uso que conforman la aplicación:

- Validar, la validación requiere exclusividad del acceso a la aplicación y esta se encuentra dictada por el Servidor Web, que es el responsable de entablar la relación de Cliente/Servidor. Debido a los requerimientos de la solución se presentan tres modalidades de usuarios para el acceso a la aplicación siendo estos: Administrador, Invitado y Consultor.
- Consulta de datos, esta tarea está limitada para uso del Administrador y del Consultor.
- Los invitados solo pueden tener acceso de consulta a datos del proyecto y a la ayuda del sistema para poder solicitar el ingreso a la aplicación.
- El acceso a la modificación de datos es exclusiva del administrador del sistema y del consultor designado.
- El manejo de las cuentas, es una tarea exclusiva del administrador del sistema.
- Impresión, este uso se implementará sólo para los casos de los informes finales de cuentas de usuarios y del resultado final del análisis de una presa, río, lago o entidad que en ese momento este siendo analizada por el usuario.
- La obtención de datos, se proporcionarán los datos en un formato de texto con formato, con el fin de dar al usuario el resultado del análisis de la entidad en cuestión.

### 2.4.1.2 JUSTIFICACIÓN DEL MODELO DE CASO DE USO

Los usos que se presentan son los requeridos para el manejo de datos y cuentas que interfieren con el sistema, por lo que se proporciona a los usuarios tareas limitadas a sus cuentas de usuario, teniendo en cuenta la seguridad de los datos del proyecto de aguas residuales. El modelo nos permite presenciar el orden de las tareas a desarrollar en la aplicación, así como proporcionar las funciones necesarias para llevar a cabo un control, uso y análisis de las muestras de aguas residuales.

Además el modelo nos permite tener los siguientes elementos:

- Documentación de las tareas disponibles en la aplicación
- Simplificación de actividades a desarrollar
- Proporciona una modulación del sistema
- Muestra una visión general del uso de la aplicación
- Modelo de Implementación, Modelo Web

### 2.4.2 MODELO DE DISEÑO

El diseño de la aplicación se desarrollara en aspectos gráficos para páginas Web, cada objeto funcional para el usuario final será presentado en paginas Web, las cuales

contendrán las funciones de administración, consulta, entre otras para el manejo de datos y de usuarios, de acuerdo a los requerimientos y orden de tareas para cada usuario o administrador.

El modo de diseño se centra en módulos generales, que agrupan las tareas siguientes:

- Módulo de Manejo de cuentas de usuarios
- Módulo de Manejo de datos de Azolves
- Módulo de Ayuda
- Módulo de Consulta de información

### 2.4.2.1 JUSTIFICACIÓN DEL MODELO DE DISEÑO

El uso de modulación de la aplicación, simplificará las tareas de desarrollo y permite para las tareas de administración y mantenimiento del desarrollo, realizar cambios en las funciones de cada una de las partes que componen la solución Web.

Permite el desarrollo de la aplicación en forma paralela, ya que cada uno de los módulos son funcionales e independientes uno de otro.

### 2.4.3 MODELO DE IMPLANTACIÓN

El modelo de implantación nos presenta la forma en cómo la solución Web se instalará en su forma lógica, mostrando las relaciones del servidor, cliente, aplicaciones y reglas a seguir para lograr que el sistema tenga un buen desempeño de acuerdo a las necesidades de la aplicación. Por lo que a continuación se listan las clases que intervienen en la solución Web:

- Implementación de un esquema de trabajo de Cliente/Servidor, bajo el aspecto de un entorno Web.
- Acceso del cliente por medio de una red, con el empleo de un Navegador Web, bajo las características de ambiente Web, protocolo de aplicación HTTP y protocolo de comunicación TCP/IP, así como sus características de seguridad, validación, enlace, y alcance geográfico.
- Servidor de aplicación Web, proporcionará el acceso directo a los clientes por medio del protocolo de aplicación HTTP, así como elemento de enlace entre los datos del proyecto de aguas residuales y la lógica de su consulta , bajo sus características de seguridad, redundancia, disponibilidad y servicios de comunicación en un entorno de Intranet y Internet.
- Servidor de Base de Datos, el cual proporciona el sistema de alojamiento de los datos del proyecto de aguas residuales.
- Servidor de Framework, este sistema proporciona el enlace entre el sistema de Web y el de Base de Datos con el fin de proporcionar soporte para la generación de páginas dinámicas y establecer una plataforma disponible para nuevas tecnologías y adecuaciones del sistema para futuras modificaciones y mantenimiento.

- Normas y lógica de aplicación, estas se encargan de generar las relaciones lógicas para la consulta, control y manejo de la aplicación en general.

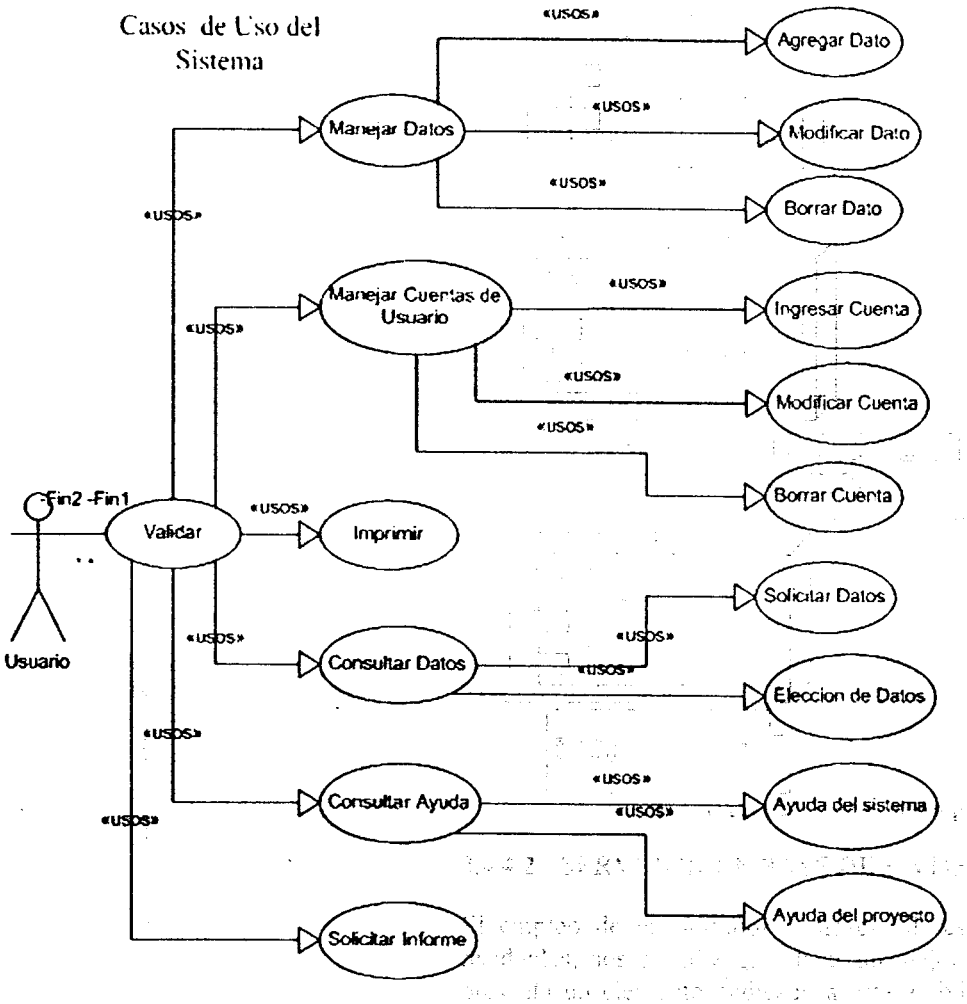



**Figura 2-3** Modelo gráfico para el Caso de Uso de la aplicación.

## 2.4.4 JUSTIFICACION DEL MODELO DE IMPLANTACIÓN

El modelo basado en Web nos permite establecer una aplicación distribuida que garantiza las siguientes cualidades:

- El esquema cliente servidor permite tener una administración centralizada de la aplicación, lo que garantiza una seguridad de los datos, mantenimiento factible, consistencia de datos y crecimiento de la aplicación.
- Alto alcance geográfico.
- Alta disponibilidad de la aplicación.
- Alto rendimiento.

- Bajo costo de implementación, en cuanto a equipo servidor y cliente.
- Fácil integración a otros sistemas.
- Acceso controlado y administrado.

El acceso de los clientes bajo un protocolo de comunicación TCP/IP y un protocolo de aplicación HTTP, nos proporcionan una alta disponibilidad del sistema, sin importar la posición geográfica de nuestros clientes, así como un ahorro en el sistema de comunicaciones que emplea un protocolo libre para su enlace. Los clientes basados en Web emplean para el uso de la aplicación solo un Navegador Web, siendo ejemplos de estos Microsoft Internet Explorer Versión 5 o superior, o Netscape con soporte de validación con Kerberos (debido a la validación y seguridad requerida por el servidor Web). Además que un navegador Web no requiere de una inversión para su instalación, permitiendo el uso desde cualquier equipo de cómputo que cuente con este componente.

### 2.4.4.1 SERVIDOR WEB (IIS VERSIÓN 5)

El servidor Web seleccionado es Microsoft Internet Information Services (IIS) Versión 5, debido a que proporciona las cualidades necesarias para poder implementar la solución Web, siendo estas las siguientes:

- Alta integración con aplicaciones Web (Framework)
- Proporciona alto rendimiento a un gran número de peticiones.
- Soporte para ambientes de alta disponibilidad de aplicaciones.
- Soporte de plataformas de desarrollo.
- Bajo costo de implementación.
- Bajo costo en equipo de servidor.
- Alta escalabilidad para las nuevas tecnologías.
- Curva de aprendizaje corta.
- Alta seguridad y control de validación a redes corporativas.
- Buen desempeño para Intranet y para Internet
- Plataforma disponible para aceptar SSL.
- Ha mostrado en el mercado Web ser un Servidor de alta disponibilidad y confiable como muestran análisis realizados por terceros (Ver notas Anexos)

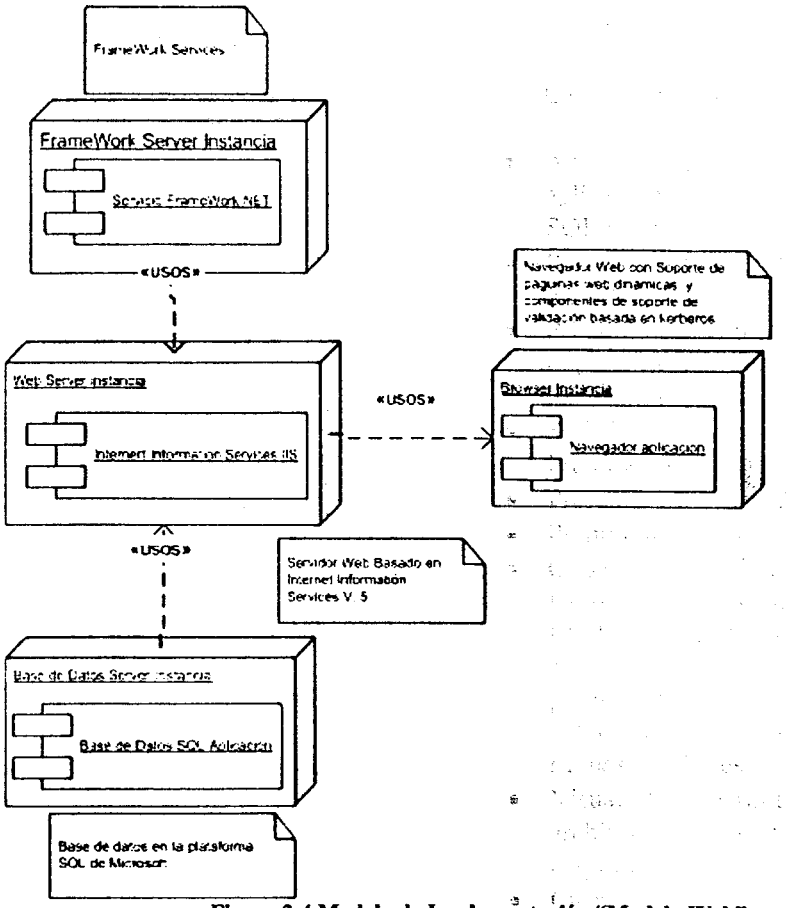



**Figura 2-4 Modelo de Implementación "Modelo Web"**

### 2.4.4.2 SERVIDOR DE BASE DE DATOS (SQL SERVER DE MICROSOFT)

El empleo de un servidor dedicado al resguardo de los datos del proyecto de aguas residuales, nos permite garantizar un mejor desempeño en la aplicación debido a que se presenta un elemento dedicado a esta actividad, permitiendo proporcionar un sistema de calidad al cliente, así como el control y centralización de los datos y administración del mismo con las tareas típicas de un DBA, y demás características que nos proporciona una base de datos.

El sistema seleccionado es Microsoft SQL Server, esta selección se llevó a cabo con base a las consideraciones de sus cualidades y disponibilidad dentro del acuerdo del proyecto de aguas residuales entre el Gobierno de la Ciudad de México y la Coordinación de Ingeniería Ambiental del Instituto de Ingeniería de la UNAM. Así como las características que nos proporciona SQL Server, siendo estas las siguientes:

- Alta integración para los sistemas basados en Web
- Completa integración con los sistemas de seguridad del servidor Web IIS, así como su sistema de comunicación y enlace.

- Soporte para desarrolladores en distintos lenguajes de programación y técnicas de desarrollo.
- Ha demostrado en análisis de terceros ser una base de datos robusta para sistemas de almacenamiento y dinamismo de datos en el Web (ver anexos).
- Proporciona un sistema de Administración Centralizado, que permite tareas típicas de un DBA.
- Alta escalabilidad y compatibilidad con sistemas existentes.
- Soporte para aplicaciones de alta disponibilidad.
- Proporciona la opción de seguridad nativa.
- El costo-beneficio que proporciona garantiza al cliente una alta disponibilidad de sus aplicaciones, respaldo y seguridad.

#### 2.4.4.3 PLATAFORMA .NET FRAMEWORK DE MICROSOFT

El .NET Framework de Microsoft, es una plataforma para desarrollar, desplegar y ejecutar servicios WEB y aplicaciones. Proporciona una ambiente productivo de alto desempeño, basado en estándares, un ambiente de multilenguajes que se integrarán a las herramientas y servicios de siguiente generación, así como la agilidad para solucionar los desafíos del despliegue y de la operación de aplicaciones Web de alta escalabilidad. El .NET Framework consiste de tres componentes principales: el "common language runtime (CLR)" proporciona un ambiente de ejecución administrado y seguro, un sistema jerárquico de bibliotecas unificadas de clases, y un componente de soporte a la versión de las páginas activas del servidor llamadas ASP.NET. Con el uso del .NET Framework se presentan las siguientes ventajas:

- Económico
- Fácil de implementar
- Alta disponibilidad
- Proporciona una base para el crecimiento de otras plataformas de desarrollo en el ambiente Web
- Alta integración con servicios y sistemas de servidores.
- Permite la implementación e integración de diversos lenguajes de programación y servicios.
- Aloja un sistema de seguridad de aplicaciones dirigido a los clientes.

### 2.4.5 LENGUAJE DE DESARROLLO

Para el desarrollo de la Solución Web se ha empleado el uso del lenguaje de programación Visual Basic, así como componentes en C# debido a la experiencia de los desarrolladores, así como elementos del lenguaje enfocado a SQL, esto permite satisfacer los requerimientos para el desarrollo de la aplicación siendo los puntos a cubrir los siguientes:

- C# en sí, es un lenguaje de programación orientado a objetos, capaz de crear y utilizar componentes COM+ y DLL, así como Servicios y aplicaciones Windows con la misma fiabilidad y rapidez que diseñar y crear aplicaciones Web y aplicaciones para dispositivos móviles. Como característica adicional, C# no

soporta los punteros como los conocemos en C++, sino que acepta y usa referencias a punteros que permiten el acceso a memoria de forma controlada. Mientras que en C++ se utilizan punteros para acceder a una dirección de memoria, en C# se utilizan referencias para acceder al identificador de un objeto.

- Alta integración entre los sistemas de control y base de datos seleccionados para la aplicación, debido a sus nativas funciones para el manejo de datos de un servidor SQL y otros.
- Presentar un sistema desarrollado capaz de ser controlado y orientado desde una metodología de Ingeniería de Software
- Proporcionar un producto documentado con el fin de cumplir con los requerimientos del proyecto y poder realizar modificaciones y actualizaciones de forma más económica para cuestiones de software y hardware.
- Satisface los requerimientos para el desarrollo de una aplicación con el apoyo del Framework ya que es de los lenguajes soportados.
- Presenta una Curva de aprendizaje corta.
- Permite la implementación de los modelos de análisis del sistema a implementar.
- C# es un lenguaje case-sensitive, es decir, sensitivo al uso de letras mayúsculas y letras minúsculas, algo que no ocurre con Visual Basic.
- El desarrollo de la aplicación puede ser llevado a cabo en otro lenguaje de programación como Visual Basic, Java, pero las características nativas de C# permiten tener un mejor control y desempeño en las funciones para el Servicio Web debido a que cuenta con simplificación de funciones y de procesos por medio de rutinas auxiliares.
- Visual Basic permite el desarrollo de la aplicación, pero el resultado final de los archivos muestra un tamaño mayor a los que proporciona C#, además de un empleo mayor de recursos por parte del servidor.
- C# permite una migración más simplificada a otras plataformas Web o Sistemas operativos, haciendo esta aplicación más portátil para futuras modificaciones o cambios de software o tecnología de hardware.

El empleo de Java para el desarrollo permite el desarrollo de la aplicación, pero el control y flujo de datos con el acceso a la base de datos no muestra un control muy cercano a los que permite SQL Server y IIS, por lo que es necesario recurrir a mayores rutinas de acceso, control, formas, y código que demandaría mayor tiempo de compilación, y recursos del servidor de producción, por lo que presentaría un bajo rendimiento en el acceso a la aplicación.

### 2.4.6 DIAGRAMA DE ACTIVIDADES

El Modelo del Diagrama de Actividades nos permite mostrar la secuencia lógica del flujo de actividades realizadas para poder obtener un resultado deseado de la aplicación, dentro de ellas podemos observar los siguientes elementos y su orden lógico:

- Inicio de Sesión
- Validación del Usuario del sistema

- Manejo de Datos, Manejo Cuentas, Administración, Consulta, impresión, entre otras actividades que pueden ser realizadas como tareas concurrentes en el sistema
- Terminar Sesión en el sistema
- Las clases resultantes para la generación de la aplicación son las siguientes:
- Clase de validación
- Clase de administración de cuentas
- Clase de administración de datos
- Clase de ayuda
- Clase de consulta de datos
- Clase de impresión
- Clase de fin de sesión

Cada una de estas clases se empleará para la generación de la aplicación debido al tamaño de cada clase y el alcance de estas dentro de la aplicación, permitiendo la simplificación de la implementación y distribución de actividades para los programadores y grupos de trabajo.

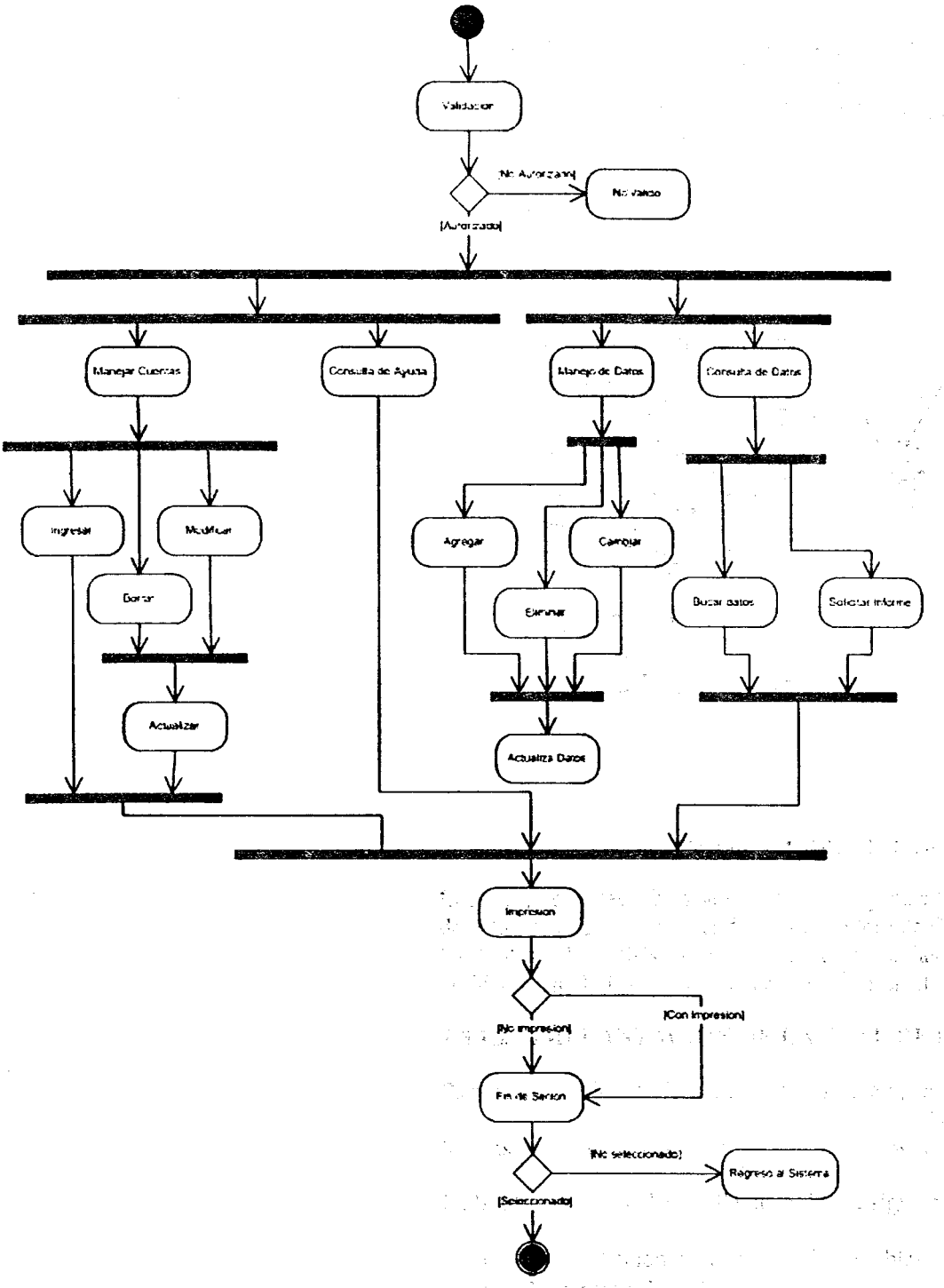



**Figura 2-5 Modelo de flujo de actividades**

## 2.5 MODELADO DE LA BASE DE DATOS

El modelado de la base de datos se llevara bajo el estereotipo de modelado relacional donde los siguientes elementos son las características para el análisis de cada muestra de estudio.

La distribución de las tablas que componen la base de datos se encuentra ordenadas en función de la naturaleza del proceso de estudio y de las pruebas que se realizan por cada una de las muestras, tomando en consideraciones los siguientes:

- **BTEX,** prueba que se lleva a cabo en el laboratorio que permite saber la cantidad de algunos compuestos.
- **Clases,** describe los tipos de lugares donde se extrae la muestra a analizar.
- **CRETIB,** es una prueba realizada en el laboratorio que emplea técnicas para determinar algunas características de elementos de carácter propias de la prueba.
- **Lugares,** contiene información de la localización de la Clase.
- **Metales_Pesados,** es el resultado de la lectura de cantidades de algunos metales pesados que se encuentran alojados en la muestra.
- **Muestras,** es la caracterización de cada una de las muestras a estudiar de cada clase.
- **Parametros_Fisicos,** es la recopilación de algunos parámetros físicos que son medidos en el laboratorio, con el fin de derminar la caracterización de la muestra.
- **Parametros_MB,** es la concentración de la lectura de elementos Microbiológicos de cada muestra de estudio, donde se ayuda a la clasificación de cada una de las clases.
- **Parametros_Quimicos,** cada una de las muestras pasa esta prueba y es reclasificada con el fin de determinar en base a los compuetos y caracteristicas químicas su naturaleza y su reacción a distintos parámetros establecidos por normas sanitarias.
- **Parametros_Fis_Plantas,** recopilación de datos físicos de las plantas.
- **Parametros_Quim_Plantas,** recopilación de datos químicos de las plantas.

Las cuales se encuentran compuestas de variables pertenecientes a cada tipo de característica, permitiendo un orden de acuerdo a las necesidades del algoritmo establecido para el análisis.

### 2.5.1 MODELO RELACIONAL

La relación de las tablas descritas nos lleva a la siguiente agrupación y modelado, de acuerdo a las necesidades de consulta, acceso, búsqueda, ingreso o modificación de los datos.

#### 2.5.1.1 MODELO ORM

El análisis de las tablas nos lleva a formular el siguiente esquema de relación de objetos, empleando la técnica de Modelado de Objetos Relacionado, siendo el siguiente esquema el resultado del análisis.

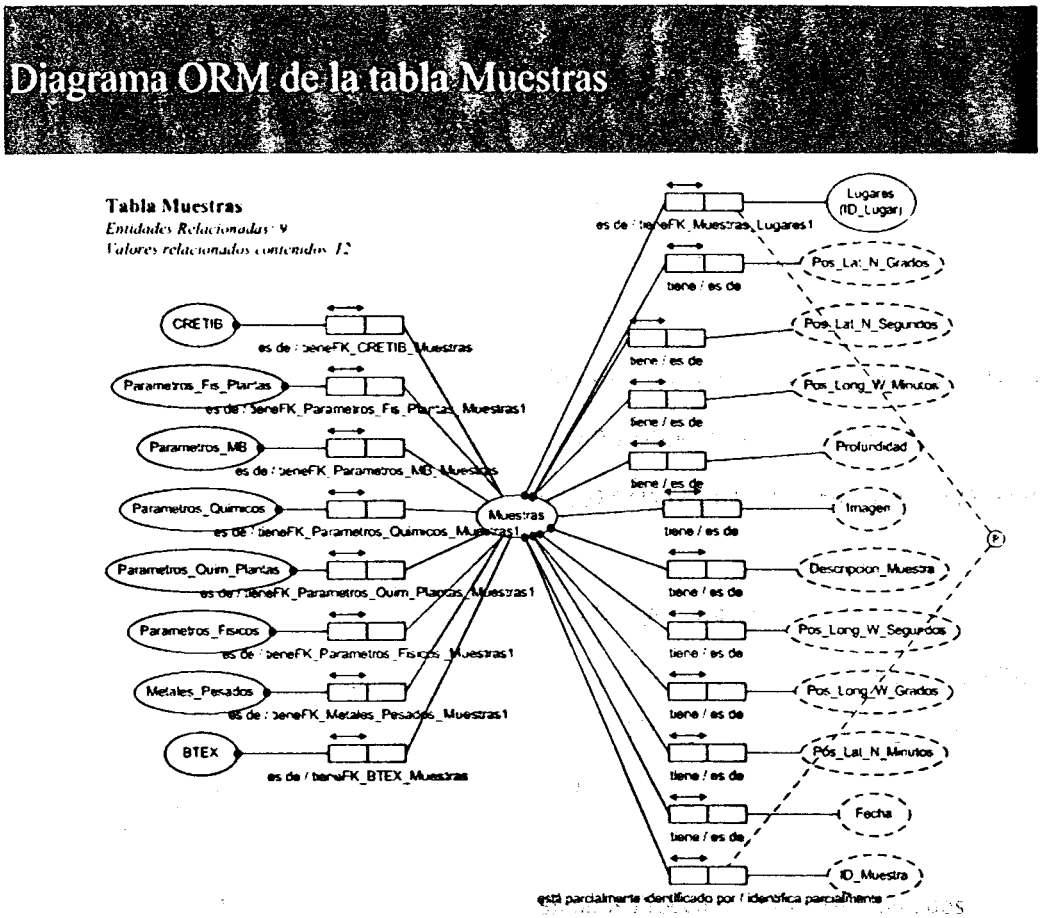



**Figura 2-6 Modelo relacional de la tabla Muestras.**

Los campos de las tablas se encuentran organizados de acuerdo a la naturaleza de su origen de estudio o tipo, manteniendo en su modelo de ORM el siguiente análisis para cada una de las tablas, el análisis de cada una de las tablas o clases mencionadas que componen la entidad Base de Datos, se pueden consultar en el Anexo.

### 2.5.1.2 IMPLANTACIÓN DE LA BASE DE DATOS

Para la formación de la base de datos se emplea el Lenguaje SQL, para la generación del Script que será posteriormente consultado para la generación de las tablas y sus relaciones lógicas de llaves primarias, foráneas, procesos, y tipo de relación.

La base de datos se desarrolla de la forma siguiente:

- Comprobación de la existencia de tablas
- Generación de tablas
- Asignación del tipo de dato

- Llaves primarias y foráneas
- Procesos almacenados
- Reglas

El proceso lógico de la base de datos para SQL Server nos lleva a los siguientes pasos:

- Creación de la base datos en SQL Server
- Creación de Tablas, relaciones y procesos en la base de datos
- Creación del Diagrama relacional

### 2.5.1.3 ENTERPRISE MANAGER

La creación de la base de datos se lleva a cabo con la consola Enterprise Manager, donde se establecen los siguientes elementos:

- Creación de la Base de Datos
- Asignación de localidad del Archivo de datos
- Asignación de localidad del Archivo Log
- Asignación de permisos

### 2.5.1.4 QUERY ANALYZER

La herramienta Query Analyzer permite la edición del Script para creación de las tablas que contendrá la base de datos, o se puede emplear cualquier otro editor de texto.

La generación del Script para las tablas emplea los siguientes elementos y estructura:

- Creación de tablas
- Relaciones, llaves o restricciones
- Relaciones bajo llaves foráneas
- Procedimientos almacenados

Cada uno de estos elementos son empleados para la generación del Script para la implementación de la base de datos con sus respectivas tablas, relaciones y procedimientos.

Las siguientes Sintaxis son las empleadas para la edición del Script:

Sintaxis de creación de tablas en SQL:

*CREATE TABLE [nombre_de_tabla] (*

*[Nombre_de_campo] [tipo_de_dato] [longitud] [NULL/NOT NULL]*

*)*

SINTAXIS para añadir relaciones, llaves o restricciones:

*ALTER TABLE [nombre_de_tabla]*

*ADD CONSTRAINT [nombre_de_restriccion] [tipo_de_restriccion]*

*[CLUSTERED/NON CLUSTERED]*

*(*

*campo_afectado*

*)*

Sintaxis para crear relaciones mediante llaves foráneas:

*ALTER TABLE [Nombre_tabla_hija]*

*ADD CONSTRAINT Nombre_relacion FOREIGN KEY*

*(*

*Llave_foranea_en_tabla_hija*

*) REFERENCES [Nombre_tabla_padre] (*

*Llave_primaria_en_tabla_padre*

*)*

Sintaxis Procedimientos Almacenados

*CREATE PROCEDURE Nombre_procedimiento*

*[declaracion* de varibales]

*AS*

*[instrucciones DML]*

### 2.5.2 SCRIPT PARA LA BASE DE DATOS

El script desarrollado para la base de datos se encuentra en el Anexo

Este script señala la generación de las tablas, sus valores, entidades, llaves primarias, llaves foráneas, y las relaciones que se requieren de acuerdo al análisis relacional de los objetos que componen cada una de las tablas y sus relaciones. Así mismo refleja la estructura que se requiere implementar de acuerdo al modelo relacional descrito.

### 2.5.2.1 DIAGRAMA A ANALIZAR

El resultado de la generación de las tablas muestra el siguiente esquema relacional de tablas, permitiendo observar las relaciones y sus características

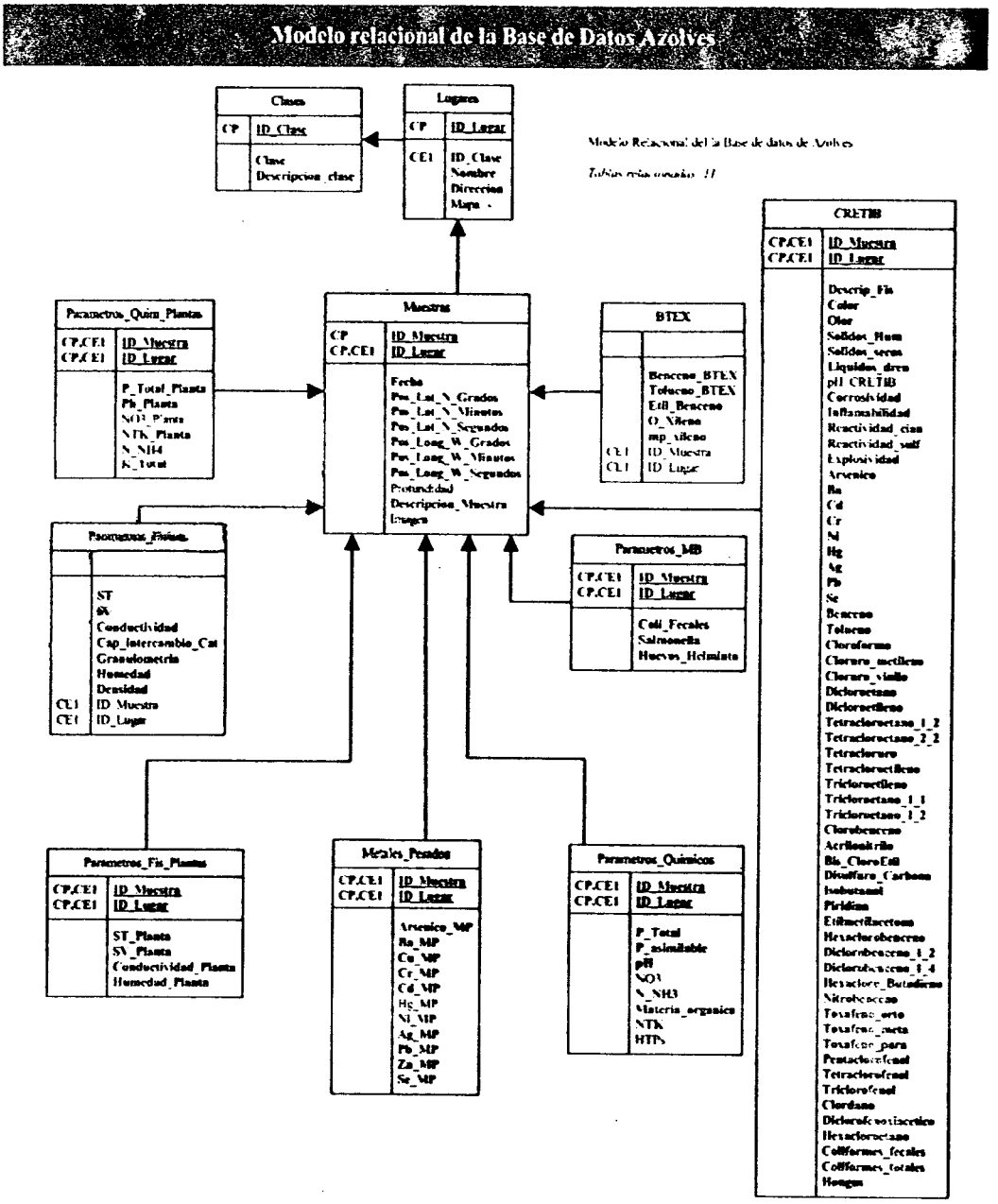



**Figura 2-7 Modelo relacional de la base de datos**

El resultado de este modelo, es basado en las relaciones que se establecen entre los objetos, las tablas, relaciones entre ellas, relaciones de confianza y al proceso de normalización para el funcionamiento de los procesos de consulta, actualización, ingreso de datos entre otros.

Este resultado refleja el orden de las clases de las tablas, donde por diseño de orden y de estructura de contenido alojan datos de acuerdo a una distribución de datos y origen de datos de laboratorio.

## 2.6 EJECUCIÓN DE PRUEBAS

La ejecución de pruebas nos permite realizar una comparación de los resultados deseados por los usuarios finales. Para estas pruebas empleamos datos reales de la aplicación, entre ellos tenemos, "Recolección de muestras", "Resultados de análisis practicados en laboratorio", "Resultados de análisis en sitio", "Datos generales sobre presas, lagos, entre otros". Este proceso automatizado surge de la labor de análisis de un solo sitio y de la recopilación de información en base a una hoja de cálculo que contiene todos los datos en varias tablas, y agrupaciones de acuerdo a la naturaleza de estudio y de criterios de manejo de datos por parte de los investigadores el Área de ingeniería Ambiental del Instituto de Ingeniería de la UNAM.

A continuación presentamos una fracción de la tabla que contiene los datos, tal y como fueron suministrados originalmente por el proyecto.

Recolección de datos de muestreo (El conjunto original de datos se puede consultar en el Anexo)

**Tabla 2-8 Recolección de datos de Muestreo**

| CARACTERISTICAS DE LA MUESTRA | | | |
|---|---|---|---|
| Clave de muestra | Fecha de muestreo | Posición geográfica | Profundidades de muestreo en presas y lagunas |
| TX-1 | 28/02/2020 | 19°19' 44.7" Lat.S. 99°13'36.6"Long.E | 0.50 m |
| TX-2 | 28/02/2020 | 19°19' 44.7" Lat.S. 99°13'36.6"Long.E | 2.40 m |
| TX-3 | 28/02/2020 | 19°19' 54.6" Lat.S. 99°13'37.3"Long.E | 0.50 m |

Se puede observar en esta hoja de cálculo la recopilación de datos de distinto tipo, carácter, o de cualidad física, química entre otros. Para efectos de pruebas de la base de datos llevaremos a cabo el llenado de datos en algunas tablas y de la consulta misma de algunos valores de interés.

### 2.6.1 PROCEDIMIENTOS ALMACENADOS.

Esta prueba permite insertar algunos datos en la tabla Clases:

*EXEC ClasesInsertar 'Planta_Tratamiento', 'Una planta de tratamiento'*

*EXEC ClasesInsertar 'Presa', 'Una presa es un lugar...'*

*EXEC ClasesInsertar 'Lagunas_Vasos_Regula', 'Una laguna y un vaso de regulacion es..'*

*EXEC ClasesInsertar 'Cauces_y_rios', 'Un cauce o un rio...'*

*EXEC ClasesInsertar 'Estaciones_Transf', 'Una estacion de transferencia...'*

En estas líneas se lleva a cabo el almacenamiento en los registros de la tabla Clases, siendo estos valores los del tipo de elemento y una descripción del mismo, dentro del proyecto se pueden ver por ejemplo, presas, plantas de tratamiento, lagunas, ríos, entre otros.

### 2.6.2 CONSULTA A LA TABLA CLASES.

En esta prueba se realiza una lectura de los datos almacenados previamente

*Select * from Clases*

| ID_Clase | Clase | Descripcion_clase |
|---|---|---|
| 1 | Planta_Tratamiento | Una planta de tratamiento |
| 2 | Presa | Una presa es un lugar... |
| 3 | Lagunas_Vasos_Regula | Una laguna y un vaso de regulación es.. |
| 4 | Cauces_y_rios | Un cauce o un rio... |
| 5 | Estaciones_Transf | Una estación de transferencia... |

Esta consulta permite obtener todos los valores que se encuentran en la tabla Clases, lo que verifica el llenado de los datos en la tabla correspondiente.

La estructura de la aplicación permite no solo el alojamiento de datos y consulta, sino que permite una estructura que es posible para futuras aplicaciones realizar nuevas o distintas tareas con los datos almacenados, ya que se cuenta con una estructura de datos básica que permite el manejo de datos por clases o grupos muy definidos.

### 2.6.3 RESULTADOS

En el Anexo se presentan más pruebas de inserción y consulta a la base de datos, donde para efectos de análisis de pruebas se ha empleado una porción de datos reales, con el fin de probar la base de datos en su capacidad de datos, tipo de datos, formato, obteniéndose los siguientes resultados:

- El llenado de los datos se puede realizar de forma correcta
- El acceso a los datos es satisfactorio para el control de datos
- El modelo implementado satisface el manejo de datos en la base datos

- El flujo de datos entre las relaciones entabladas permite el acceso a datos por medio de las referencias de consulta ya establecidas
- Permite el modelo establecido realizar cambios en su contenido, con el fin de actualizar o de ingresar nuevas clases que almacenen datos, sin tener un impacto considerable en el desarrollo de la aplicación o funcionamiento

# CAPÍTULO 3 IMPLEMENTACIÓN DE LA SEGURIDAD

La seguridad es un tema que hoy en día debe ser considerado como parte integral del diseño de una aplicación basada en Web y debe ocupar un lugar importante durante la planeación e implementación de cualquier componente. Cuando una aplicación se instala en los servidores de producción, existe un riesgo permanente a la seguridad de la misma y del resto de la infraestructura. Una aplicación que ha sido víctima de un ataque a su seguridad, implica una pérdida económica importante, no solo por el tiempo que se invierte en corregir el problema y verificar la validez de la información que se almacena; además implica la perdida de la confianza de los clientes.

Los riesgos de seguridad no se encuentran directamente relacionados con una plataforma o tecnología, empleada en la implementación y soporte de una aplicación basada en Web. Es cada vez más común escuchar en los seminarios de seguridad en cómputo que no existe una plataforma o aplicación exenta de fallas en su seguridad. El tema de la seguridad es hoy en día uno de los tópicos que más atención recibe por parte de los medios de comunicación y por consiguiente de los clientes. Las empresas responsables invierten recursos económicos en sostener personal dedicado a garantizar la seguridad y corregir cualquier problema que se presente.

En este capitulo se exponen los diferentes conceptos y recomendaciones usados en la planeación de la seguridad, así mismo se presenta la filosofía empleada en el diseño de la seguridad implementada en la solución Web.

## 3.1 CONSIDERACIONES SOBRE LA SEGURIDAD

La creación de aplicaciones Web distribuidas seguras constituye un verdadero desafío. El alcance de la seguridad de una aplicación lo define su punto más débil. Las aplicaciones distribuidas contienen muchas piezas móviles y conseguir que esas piezas funcionen conjuntamente de forma segura requiere un conocimiento de base de numerosos productos y tecnologías.

### 3.1.1 VALOR DE LA INFORMACIÓN

La información es un conjunto de datos, que han sido procesados para generar una forma significativa para el receptor, el valor contenido en la información, cambia en función de las necesidades del receptor, del tiempo de vida de la información y más significativamente en la confianza que deposite sobre el estado que guarda la información. En este último punto, es donde se requieren esquemas y medidas de seguridad que resguarden el estado de la información, permitiendo que el receptor confíe en la veracidad de la información que recibe.

La seguridad es un factor clave para los arquitectos y los desarrolladores de aplicaciones, las aplicaciones que almacenan información confidencial deben adoptar medidas de protección frente a ataques malintencionados y competidores directos. Los esquemas de seguridad buscan proteger cuatro elementos fundamentales:

- **La Integridad:** confirmación de que los datos que han sido enviados, recibidos o almacenados son completos y no han sido modificados. La integridad es especialmente importante en relación con la autenticación para la conclusión de transacciones o en los casos en los que la exactitud de los datos es crítica
- **La Confiabilidad:** significa que los datos son accesibles, inclusive en casos de alteraciones, cortes de corriente, catástrofes naturales, accidentes o ataques. Esta característica es particularmente importante cuando una avería de la red puede provocar interrupciones o reacciones en cadena que afecten las operaciones de la empresa.
- **La Autenticación:** confirmación de la identidad declarada de usuarios. Son necesarios métodos de autenticación adecuados para muchos servicios y aplicaciones.
- **La Confidencialidad**: protección de las comunicaciones o los datos almacenados contra su interceptación y lectura por parte de personas no autorizadas. La confidencialidad es necesaria para la transmisión de datos sensibles y es uno de los requisitos principales a la hora de dar respuesta a las inquietudes en materia de intimidad de los usuarios de las redes de comunicación.

A la hora de diseñar un modelo de seguridad para una aplicación, es necesario considerar los requisitos de seguridad desde una perspectiva empresarial y las implicaciones que el modelo seleccionado pueda tener en el rendimiento, la escalabilidad y la distribución.

En una aplicación basada en Web, durante la etapa de diseño se recomienda contar con un apartado donde se traten las cuestiones sobre seguridad. Se deben tener en cuenta y posiblemente abordar los puntos de la Tabla 3-1, en ella Kercher Jeff, (2002) expone de manera resumida los puntos básicos para iniciar la planeación de un esquema de seguridad adecuado dentro del modelo de desarrollo de cualquier aplicación.

**Tabla 3-1 Puntos a evaluar durante la planeación de una aplicación basada en Web.**

| Concepto | Descripción |
|---|---|
| Objetivos de seguridad. | Comprender qué es lo que se está protegiendo y asegurarse de poder describirlo. |
| Riesgos de seguridad. | Comprender la vulnerabilidad de la aplicación. Asimismo, se debe tener presente la importancia de los riesgos potenciales relacionados con su empresa. |
| Autenticación. | Se trata del proceso de aceptación de las credenciales de un usuario y la validación de las mismas frente a una autoridad designada. La identidad del usuario (o potencialmente, la de una aplicación o equipo) se denomina principal de seguridad. El cliente debe proporcionar las credenciales para permitir que el servidor compruebe la identidad del principal. Una vez conocida la identidad, la aplicación podrá autorizar al principal para que tenga acceso a los recursos del sistema. A continuación se proporcionan distintos criterios que servirán de ayuda a la hora de seleccionar el mecanismo de autenticación adecuado. |
| Autorización. | En este proceso se determina si a la identidad comprobada se le permite tener acceso a un recurso específico. |
| Seguridad en la transmisión de datos. | Mediante el cifrado de los datos y a medida que se desplazan por la red, se puede asegurar que no se consulten ni se alteren durante su transmisión. Se debe considerar el grado de seguridad que precisan los datos durante la transmisión. |
| Representación. | Este mecanismo permite que se ejecute un proceso de servidor empleando las credenciales de seguridad del cliente. Cuando el servidor representa al cliente, cualquier operación realizada por el servidor se efectúa utilizando las credenciales del cliente. La representación no permite que el servidor tenga acceso a los recursos remotos en nombre del cliente. Esta operación requiere el mecanismo de delegación. |
| Delegación. | Al igual que sucede con la representación, este mecanismo permite que se ejecute un proceso de servidor utilizando las credenciales de seguridad del cliente. Sin embargo, la delegación resulta más eficaz y permite que el proceso de servidor realice llamadas a otros equipos mientras actúa como cliente. |
| Seguridad del sistema operativo. | Este punto hace referencia al establecimiento de las listas de control de acceso (ACL) adecuadas y a la seguridad de la red con el fin de evitar que los intrusos obtengan acceso a los recursos protegidos. Se deben establecer las listas ACL adecuadas en los recursos apropiados para permitir que únicamente los clientes indicados obtengan acceso. |
| Seguridad en el acceso físico. | Este aspecto hace referencia a la ubicación del equipo servidor en un lugar seguro. Se trata de un punto fundamental que no se debe descuidar. |
| Seguridad de acceso al código. | Permite que se establezcan distintos grados de confianza en el código, dependiendo de dónde proceda y de otros aspectos de su identidad. Se debe tener en cuenta cómo crear permisos de acceso propios. |

*-Fuente: Kercher Jeff, 2002 -*

En la actualidad es recomendable que cada empresa cuente con su modelo de seguridad, al cual deben de apegarse todas las aplicaciones que se desarrollen para la misma. Una aplicación mal planificada en cuestión de seguridad puede ser motivo de quejas y provocar un riesgo de seguridad. Las aplicaciones basadas en Web, son de alto riesgo, debido a que se encuentran expuestas a un medio no controlado como es Internet, si un sitio Web tiene un gran impacto, es porque ofrece un servicio muy requerido o pertenece a una compañía importante, recibirá cientos o miles de "hits" por minuto, no todos serán provenientes de clientes bien intencionados, los servidores Web de producción reciben diariamente cientos de intentos de ataques ejecutados por personas con el conocimiento suficiente o bien, directamente por equipos infectados por un virus o programas elaborados con esta finalidad.

Cuando un ataque contra una aplicación Web es exitoso, se ven comprometidos no solo la imagen corporativa de nuestro cliente, es seguro que el intruso haya logrado recabar información referente a los recursos físicos y lógicos de nuestro cliente, así como también de las operaciones que se realizan, esto último puede involucrar información de carácter comercial, información de negocios y datos de los clientes de las aplicaciones que se ejecutan en el servidor en cuestión.

Los servidores que han sufrido un ataque, se encuentran comprometidos en su seguridad, debido al robo de las cuentas y contraseñas del sistema, la acción de programas implantados por el intruso y cambios en sus esquemas de seguridad. De igual manera que existe una planeación de seguridad, debe de existir un esquema de recuperación de desastre que indique las acciones a seguir durante y después de un ataque informático. Un plan de recuperación correctamente formulado y probado repercutirá en una menor cantidad de inconvenientes a los consumidores del servicio y en muchos casos permiten recuperar la presencia del producto en la Web en cuestión de horas.

En los seminarios de seguridad los temas giran regularmente en torno de ataques dirigidos contra la funcionalidad de un sistema o aplicación. Esto deja un vacío importante; no todos los problemas de seguridad de una aplicación Web llegan vía Internet. Dentro de las instalaciones donde se encuentran los equipos de comunicaciones y servidores, pueden presentarse eventos que afecten directamente, fallas en el suministro eléctrico, sistemas de refrigeración no operantes, instalación no adecuada de componentes en los servidores de producción, fallas intencionadas en el hardware y fenómenos climáticos. Todos estos elementos pueden llegar a causar la pérdida total de la aplicación y de la información que contienen. Es de esperar que nadie considere una inundación o tornado como un riesgo de seguridad para una aplicación basada en Web, no así un empleado inconforme, el cual puede cortar el suministro eléctrico o dañar el almacén de datos de los servidores. De igual forma que se diseña la seguridad lógica de una aplicación, nunca está de más revisar la seguridad física del lugar donde residen los equipos que dan soporte a nuestra aplicación.

Una aplicación basada en Web no es sinónimo de inseguridad, hoy en día podemos consultar sitios Web de comprobada seguridad, que reflejan la labor de varias personas que constantemente monitorean y previenen las fallas que el sistema pueda presentar. En conjunto con esta labor, la revisión de los esquemas de seguridad, los simulacros y revisiones al programa de recuperación de desastres, garantizan la presencia de una aplicación en Internet por más tiempo.

## 3.2 PANORAMA DEL CLIENTE

La información referente a los esquemas de seguridad establecidos por nuestro cliente es nula, motivo por el cual se ha diseñado y propuesto un escenario que represente las actividades y estructura con la que cuenta nuestro cliente. Usando este escenario plantearemos los riesgos típicos que enfrentará nuestra aplicación, del estudio de estos riesgos propondremos los esquemas de seguridad. Los esquemas de seguridad resultantes tendrán como finalidad garantizar un ambiente mínimo recomendado, de fácil implementación y bajo impacto al entorno de la organización.

Como primer punto, es necesario hacer énfasis en que el apoyo por parte de la gente con el poder de decisión (cuerpo directivo, dueños de los recursos, gerencia, entre otros.) es fundamental para el éxito de un esquema de seguridad, ya que sin él, algunos elementos de dicho esquema no tendrían validez (v. gr. políticas y procedimientos).

Es vital mantener en constante capacitación tanto al personal antiguo como al nuevo mediante cursos, seminarios, congresos o cualquier otro medio. La mejor defensa es el conocimiento. Los usuarios deben conocer el uso adecuado de los sistemas de cómputo y saber cómo protegerse a sí mismos de accesos no autorizados. Debe crearse una cultura de seguridad, haciendo ver a la gente involucrada los peligros a los que se está expuesto en un ambiente tan hostil como el que ha generado la evolución de las actuales redes de computadoras.

### 3.2.1 ESCENARIO DEL CLIENTE

La necesidad de contar con una descripción clara y detallada del medio ambiente en el cual se encontrará operando una aplicación, permite diseñar y dictar las medidas de seguridad adecuadas para garantizar un nivel óptimo de seguridad sin que ello represente entregar a los usuarios sistemas complejos de validación o consumo excesivo de recursos físicos y económicos del departamento de informática. De igual forma al contar con esta información se pueden definir pruebas específicas, para determinados eventos o tipos de usuarios que tendremos durante la operación normal del producto terminado.

En el diseño del escenario fue necesario hacer uso de la poca información proporcionada por nuestro cliente, esta información fue complementada con la experiencia laboral en dependencias del gobierno por parte de profesores y compañeros de trabajo. Así mismo en la especificación de los requerimientos del sistema encontramos información adicional referente a los consumidores potenciales de la aplicación.

Para el diseño del escenario se cuido presentar una infraestructura mínima necesaria para garantizar el funcionamiento correcto de una organización, esto es: reflejar una arquitectura básica de red, incluir los elementos básicos en un centro de cómputo de propósito general, suponer el empleo de un esquema jerárquico en la distribución de los departamentos y sus recursos informáticos. Así mismo incluimos elementos que nos fue garantizado que se disponen, como es el caso de: Servidores Web independientes de otras dependencias, enlaces de comunicaciones dedicados, consumidores potenciales de la aplicación, entre otros elementos.

Durante el presente capitulo se ha mencionado en diferentes ocasiones la necesidad de contar con una infraestructura física adecuada que facilite el buen funcionamiento de los equipos críticos para la organización, estos elementos no figuran dentro de los elementos del escenario propuesto, en virtud del desconocimiento de los elementos con los que se dispone en el medio ambiente donde operará nuestra solución.

Los componentes considerados en el diseño del escenario son descritos de forma breve en la Tabla 3-2, así mismo estos elementos y su relación entre sí son expuestos de forma grafica en la Figura 3-1.

**Tabla 3-2 Descripción de los elementos involucrados en el escenario propuesto para la Secretaría del Medio Ambiente del Distrito Federal.**

| Elemento | | Descripción |
|---|---|---|
| Secretaría del Medio Ambiente del distrito Federal ("SMA-DF") | | Es nuestro cliente y principal consumidor de los servicios ofrecidos por la solución Web. |
| Ciudadanos | | Es el conjunto de clientes ocasionales, los cuales no tienen relación alguna con la "SMA-DF" |
| Usuarios Móviles | | Es todo aquello personal directo e indirecto de "SMA-DF" que por cuestiones laborales realizan acceso a los recursos en instalaciones ajenas a las de nuestro cliente o de cualquier otra dependencia gubernamental. |
| Secretarías con relación directa | | Son todas aquellas Secretaría y dependencias del gobierno que mantienen una relación de trabajo estrecha con la "SMA-DF". |
| Otras Secretarías | | Son todas aquellas Secretarías y dependencias del gobierno que no mantienen una relación de trabajo estrecha con la "SMA-DF". |
| Otras ciudades | | Son todos aquellos usuarios antes mencionados que radican en otras entidades federativas. |
| 1 | Servidores de Impresión | Servidores dedicados a la administración de las impresoras. |
| 2 | Equipos y usuarios móviles | Personal y equipo con movilidad dentro de la "SMA-DF" |
| 3 | Impresoras de red | Impresoras de gran volumen que controladas por el servidor de impresión |
| 4 | Equipos de escritorio | Conocidas como "PC", equipos de cómputo de propósito general |
| 5 | Otros equipos de cómputo | Abarca equipos de cómputo de propósito particular como lo son las estaciones de trabajo, equipos de edición de video, entre otros. |
| 6 | Servidores de archivos | Servidores dedicados a la administración de documentos. |
| 7 | Servidores de uso exclusivo | Servidores de uso particular y necesario para la organización. (v. gr. Controladores de dominio, Emisor de certificados, entre otros). |
| 8 | Servidor de aplicaciones | Servidores dedicados a la administración y ejecución de aplicaciones diversas. |
| 9 | Servidor de bases de datos | Servidores dedicados al soporte del manejador de bases de datos. |
| 10 | Servidor Web | Servidor dedicado a la publicación y exposición de servicios por medio de Internet |
| 11 | Servidor de correo | Servidor de soporte para el sistema de correo electrónico. |
| 12 | Subred | Estructura física de conexión de los equipos de cómputo. |
| 13 | Equipos de conexión interna | Equipos de red dedicados a mantener la comunicación entre las diferentes subredes (v. gr. Hub's, bridge, Switch ). |
| 14 | Equipos de conexión externa | Equipos de redes usados para mantener el enlace con Internet. |
| 15 | Banco de Modem's | Equipo de red usado para mantener conexiones por medio de la infraestructura telefónica. |
| 16 | Internet | Representado por una nube, en ella se agrupan todas las redes de comunicación de Internet. |

## 3.3 CONSUMIDORES DE LA APLICACIÓN

Los requerimientos planteados por el cliente, buscan obtener una aplicación con una interfase muy simple y de acceso desde cualquier lugar de trabajo, motivo por el cual se optó por una solución Web, en la Figura 3-1, se pueden localizar los cinco clientes principales de nuestra aplicación (e. i. Secretaría del Medio Ambiente del distrito Federal ("SMA-DF"), Ciudadanos, Usuarios Móviles, Secretarías con relación directa, Otras Secretarías, Otras ciudades), cada uno de ellos consumirá el servicio ofrecido por nuestra aplicación y presentarán su propios requerimientos y consideraciones.

### 3.3.1 SECRETARÍA DEL MEDIO AMBIENTE DEL DISTRITO FEDERAL ("SMA-DF")

Es el consumidor principal de los servicios ofrecidos por nuestra solución, sus componentes abarcan la infraestructura necesaria para el soporte de todos sus empleados en un ambiente de grupos de trabajo, con equipos conectados entre si por medio de una red de comunicaciones.

Los equipos de cómputo no son homogéneos, se pueden encontrar computadoras personales, laptops, estaciones de trabajo, terminales de trabajo, entre otros, esta diversidad se ve reflejada en el sistema operativo que se encuentra instalado, encontrando las plataformas más populares del mercado en la mayoría de sus versiones.

La infraestructura de servidores garantiza una autonomía parcial de otras dependencias gubernamentales, permitiendo al la propia dependencia su propias decisiones referentes al uso y disponibilidad de recursos por parte de una aplicación determinada. Así mismo esta independencia de otros organismos, ofrece la posibilidad de realizar las labores de mantenimiento y soporte de nuestra aplicación sin interferir con aplicaciones ajenas.

De manera similar la infraestructura de comunicaciones es parcialmente autónoma, prestando mayor atención al hecho de contar con enlaces dedicados lo que permitirá a cualquier aplicación desarrollada para esta dependencia, el contar con un sistema de comunicaciones persistente. Motivo por el cual se espera que las aplicaciones Web ha ejecutarse en este ambiente sean disponibles en cualquier momento.

Se puede ver que nuestro principal cliente cuenta con los recursos necesarios para presentar una aplicación de fácil acceso, alta disponibilidad y alto nivel de seguridad. Este último punto se puede alcanzar por medio de políticas de seguridad dirigidas al equipo de cómputo, al esquema de cuentas de usuarios y a las políticas aplicadas sobre los equipos de comunicación, dejando a nuestro cliente como un medio ambiente controlado, sobre el cual se puede tener un control óptimo.

### 3.3.2 CONSUMIDORES SECUNDARIOS

Aun cuando la aplicación no fue directamente desarrollada para este tipo de usuarios los requerimientos originales de la aplicación y los recursos físicos por parte de nuestro cliente permitieron extender el tipo de usuarios que tendrían acceso a esta solución Web. Al

margen de estos motivos, en una segunda etapa se considera la generar una mesa de trabajo en torno a la problemática que dio origen a este proyecto, motivo por el cual se planea una ampliación a esta solución Web para incrementar el valor agregado de la misma, este punto aun se encuentra en vías de aprobación.

### 3.3.2.1 PERSONA SIN RELACIÓN CON DEPENDENCIAS DE GOBIERNO

En la Figura 3-1 se puede localizar a un grupo de usuarios particulares agrupados bajo el nombre de ciudadanos, este grupo esta integrado por todos los usuarios del sistema que no cuentan con relación alguna con dependencias de gobierno, adicionalmente abarca a todo usuario sin autorización expresa para acceder a los recursos protegidos del sistema.

Dentro de la funcionalidad de la solución Web, los "Ciudadanos" son considerados para tener acceso a una sección informativa del proyecto, en este nivel de acceso se busca presentar información relacionada con la problemática de los Azolves, acciones y propuestas realizadas por la Secretaría del Medio Ambiente del Distrito Federal.

Para este tipo de clientes es difícil anticipar la versión de su sistema operativo, el tipo de navegador con el que cuenta, su tipo de conexión, entre otros aspectos. Aun cuando este tipo de usuarios no consumirán todos los recursos que ofrece la solución Web, una buena experiencia será favorable para la imagen de nuestro cliente, que a fin de cuentas debe de otorgar el mejor servicio posible a la ciudadanía.

En cuestiones de seguridad, el "Ciudadano" se presenta como un cliente sobre el cual no se puede aplicar una política de seguridad sobre su equipo, tampoco se puede predecir su origen o el contenido malicioso que exista en su equipo de cómputo.

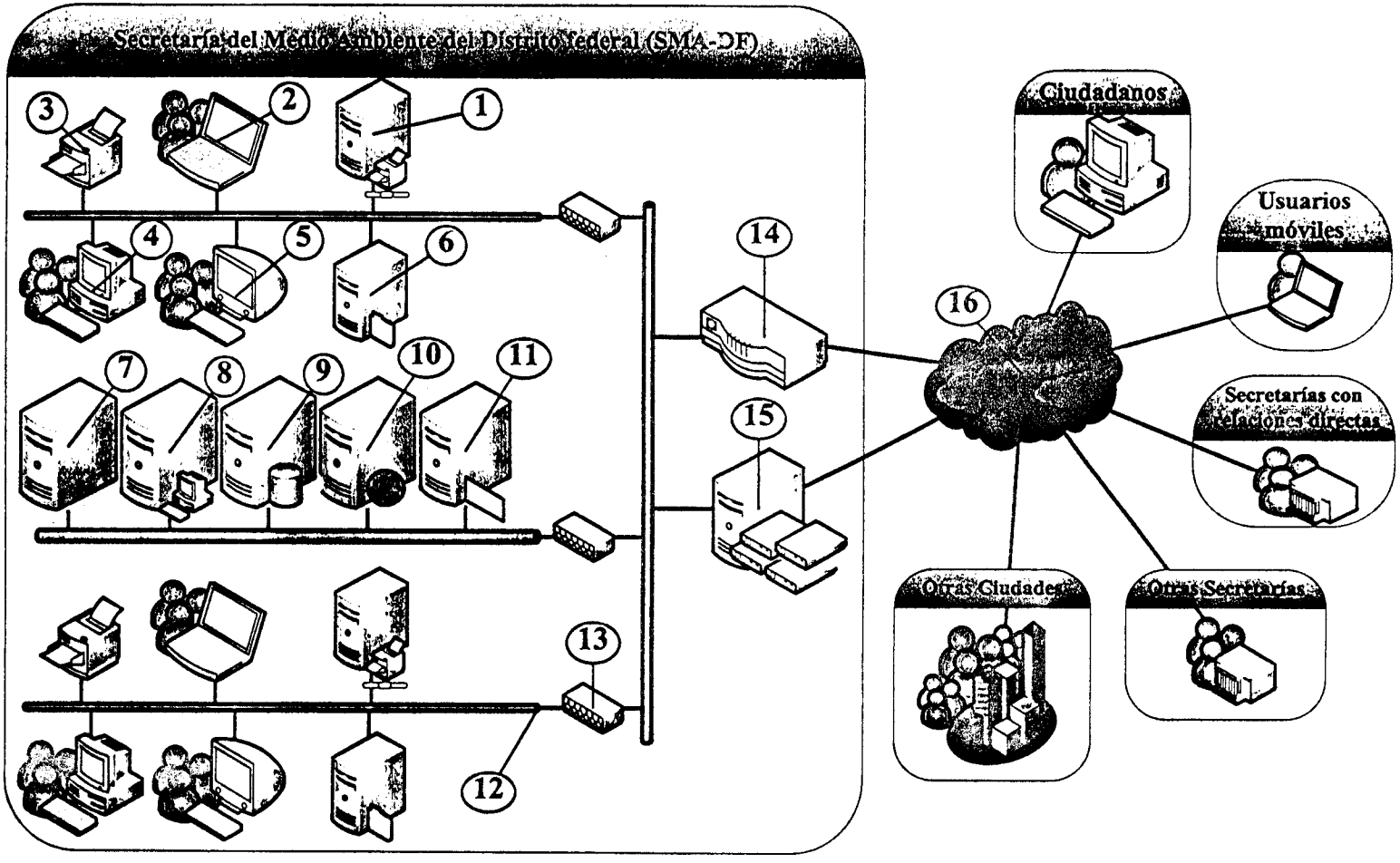



**Figura 3-1 Escenario propuesto para el entorno de operación de la Secretaría del Medio Ambiente del Distrito Federal.**

### 3.3.2.2 USUARIOS MÓVILES

Como "Usuario móvil", se agrupan a todos aquellos usuarios que cuentan con una relación directa con la Secretaría del Medio Ambiente del Distrito Federal y han sido autorizados para hacer uso de los recursos disponibles en esta solución. Entre estos usuarios se pueden encontrar: personal de directo de la Secretaría (SMA-DF), personal de otras dependencias gubernamentales, personal sin relación directa con alguna dependencia de gobierno con autorización para utilizar esta solución.

El tipo de conexión usado por estos clientes, varía en función de los recursos disponibles por cada cliente. Para el caso de los clientes gubernamentales, cuando el acceso se realiza desde alguna instalación oficial, el enlace disponible podrá ser dedicado y contar con un ancho de banda aceptable, en le caso de conexiones provenientes de instalaciones no oficiales, el tipo de enlace será muy variado.

El universo de equipo de cómputo disponible para este grupo de usuarios, es muy variado, predominando el uso de computadores personales. Al contar en este grupo con personal directo de la "SMA-DF", tenemos un sector de equipos controlado por las políticas implementadas por dicha Secretaría. Los equipos de cómputo de otras dependencias y de terceros, son una vez más elementos que escapan al control de las políticas de seguridad establecidas.

### 3.3.2.3 SECRETARÍAS CON RELACIÓN DIRECTA

Son todos aquellos usuarios del sistema que pertenecen a otras dependencias gubernamentales, que por motivos labores cuentan con una relación de trabajo directa con el personal de la Secretaría del Medio Ambiente del Distrito Federal. En este grupo de usuarios encontraremos personas a las cuales se les permite el acceso a los recursos del sistema y otras a las cuales no se les tiene contemplados como usuarios del sistema.

Las características de los equipos de cómputo y la infraestructura de comunicaciones con los que se cuenta en este grupo de usuarios es similar a la infraestructura descrita en la sección 3.3.1 del presente capitulo, motivo por el cual es posible que se cuente con políticas de seguridad similares a las implementadas por nuestro cliente, en casos muy particulares se puede llegar a un acuerdo para solicitar la aplicación de ciertas políticas que se consideren necesarias para mantener un nivel de seguridad aceptado.

### 3.3.2.4 OTRAS SECRETARÍAS

Dentro de esta categoría se agrupan a todos aquellos usuarios que mantienen una relación directa con alguna dependencia gubernamental diferente a la Secretaría del Medio Ambiente del Distrito Federal, pero que carecen de una relación de trabajo directa con la Secretaría antes mencionada. Los usuarios aquí agrupados solo cuentan con la opción de ingresar a la parte informativa del proyecto, en caso de contar con usuarios con

autorización para acceder a los recursos del sistema, entraremos en el caso desarrollado en el punto 3.3.2.3 del presente capítulo.

El tipo de equipos que esperamos serán muy similares a los expuestos en las secciones 3.3.1, 3.3.2.3 del presente capitulo, así mismo estos equipos tendrán sus propios esquemas y políticas de seguridad, a este punto debemos agregar, que existe la posibilidad de negociar acuerdos en cuanto a los requerimientos de software que deberán tener los equipos de cómputo de pertenecientes a este grupo de usuarios.

### 3.3.2.5 OTRAS CIUDADES

Este grupo hace referencia a todos los clientes secundarios de la aplicación que cuenten con la cualidad de residir en otra entidad federativa diferente a la localización de la propia Secretaría. Los riegos y requerimientos de este grupo de usuarios son idénticos a los presentados por cada uno de los grupos que conforman a los consumidores secundarios, los riesgos de seguridad son exactamente los mismos debido a que todos los clientes secundarios realizan el acceso a los recursos del sistema por medio de Internet.

## 3.4 IDENTIFICACIÓN DE LAS AMENAZAS DE LA SEGURIDAD

Las aplicaciones basadas en Web, por su naturaleza de alta disponibilidad, pueden ser blanco de diversos grupos de intrusos. Las aplicaciones Web son en su mayoría una presencia virtual de instituciones y corporaciones, alojando en sus contenidos la visión e intereses de sus partes físicas, dejando de lado a las soluciones Web dedicadas al comercio, las aplicaciones Web son blancos adecuados de instituciones rivales o personas con rencillas personales o políticas, que ven en este tipo de aplicaciones una manera de causar un daño a los bienes o imagen de una determinada compañía.

Basados en el "Security Referente Handbook" de Symantec Corporation mencionaremos las principales amenazas a la seguridad de nuestra aplicación. Las amenazas a la seguridad se pueden agrupar en tres grupos principales (Figura 3-2)

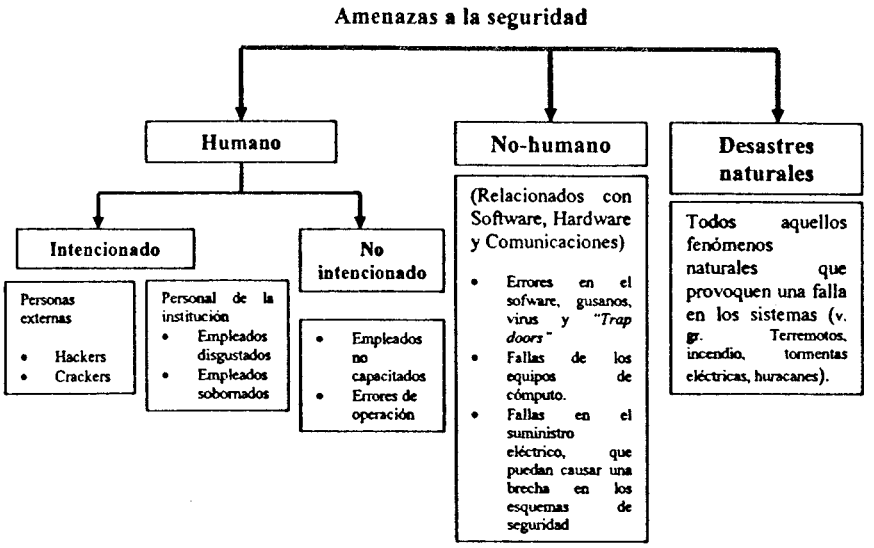



**Figura 3-2 Amenazas a la seguridad**

*Fuente: Security Reference Handbook*

Es preciso tener en cuenta todos los factores que pueden amenazar la privacidad y no solamente los intencionados. Desde el punto de vista de los usuarios, los peligros derivados de los incidentes del entorno o de errores humanos que alteren la red pueden ser tan costosos como los ataques intencionados. La seguridad de las redes y la información puede entenderse como la capacidad de las redes o de los sistemas de información para resistir, con un determinado nivel de confianza, todos los accidentes o acciones malintencionadas, que pongan en peligro la disponibilidad, autenticidad, integridad y confidencialidad de los datos almacenados o transmitidos y de los correspondientes servicios que dichas redes y sistemas ofrecen o hacen accesibles.

### 3.4.1 RIESGOS

Cada uno de los posibles actores que intervienen en los riesgos de seguridad cuentan con herramientas en mayor o menor medida especializas y buscan diversos puntos de irrupción a los sistemas. El continuo monitoreo de los sistemas que deseamos proteger, requiere del uso de técnicas y herramientas adecuadas, por tal motivo se debe de considerar la inversión en materia de actualización del software y personal dedicado a la prevención de incidentes de seguridad.

Entre las principales amenazas o riesgos que enfrentan los sistemas conectados a red son:

- **Acceso no autorizado a servidores y redes de computadoras:** el acceso no autorizado a servidores o redes de computadoras se realiza habitualmente con el fin de copiar, modificar o destruir datos. Técnicamente, se conoce como intrusión y adopta varias modalidades: explotación de información interna, ataques aprovechando la tendencia de la gente a utilizar contraseñas previsibles, aprovechar

la confianza de la gente a revelar información a personas en apariencia fiable (Ingeniería Social), e interceptación de contraseñas.

- **Accidentes no Provocados: numerosos problemas de seguridad se deben a accidentes imprevistos o no provocados como:** son tormentas, inundaciones, incendios, terremotos, interrupción del servicio por obras de construcción, defectos de programas y errores humanos o deficiencias de la gestión del operador, el proveedor de servicio o el usuario.
- **Ataque frontal:** Ataques de denegación de servicio (DoS) el cual puede dañar hasta un sistema seguro inundándolo con peticiones inapropiadas o mal formuladas que saturarían el sistema o crearían procesos que pondrían en peligro el sistema o sus datos, además de otros sistemas que comuniquen con él.
- **Búsqueda entre los datos de autenticación:** muchos métodos de autenticación por defecto en los sistemas operativos dependen de enviarle su información de autentificación "en texto plano" donde su nombre de usuario y contraseña se le envían por medio de la red en texto común o sin encriptar. Existen herramientas a disposición para quienes tengan accesos a su red (o Internet, si obtiene acceso a su sistema mientras la usa) para "husmear" o detectar su contraseña grabando todos los datos transferidos por medio de la red y examinarlos para encontrar declaraciones de inicios de sesión comunes. Este método se puede usar para encontrar cualquier información enviada sin encriptar, como pueden ser contraseñas o documentos.
- **Declaración Falsa:** a la hora de efectuar una conexión a la red o de recibir datos, el usuario formula hipótesis sobre la identidad de su interlocutor en función del contexto de la comunicación. Para la red, el mayor riesgo de ataque procede de la gente que conoce el contexto. Por tal razón, las declaraciones falsas de personas físicas o jurídicas pueden causar daños de diversos tipos. como pueden ser transmitir datos confidenciales a personas no autorizadas, rechazo de un contrato, entre otros.
- **Ejecución de Programas que Modifican y Destruyen los Datos:** Las computadoras funcionan con programas, pero lamentablemente, los programas pueden usarse también para desactivar una computadora y para borrar o modificar los datos. Cuando esto ocurre en una computadora que forma parte de una red, los efectos de estas alteraciones pueden tener un alcance considerable. Por ejemplo, un virus es un programa mal intencionado que reproduce su propio código que se adhiere, de modo que cuando se ejecuta el programa infectado se activa el código del virus.
- **Espionaje de las comunicaciones:** la comunicación puede ser interceptada y los datos copiados o modificados. La interceptación puede realizarse mediante el acceso físico a las líneas de las redes, por ejemplo, interviniendo la línea, o controlando las transmisiones.
- **Infecciones por virus:** Un virus es simplemente un programa. Una secuencia de instrucciones y rutinas creadas con el único objetivo de alterar el correcto funcionamiento del sistema y, en la inmensa mayoría de los casos, corromper o destruir parte o la totalidad de los datos almacenados en el disco.
- **Intrusión por medio de "bug's" de seguridad o de un "loophole" (rendija de seguridad):** Existen errores en el software que, si son explotados, podrían causar graves daños a un sistema no protegido. Se debe procurar el uso correcto y

constante de las herramientas que estén a disposición, como actualizaciones de paquetes de redes y alertas de seguridad, para resolver problemas de seguridad tan pronto como sean descubiertos. Por último, se debe asegurar que el sistema no tenga programas innecesarios que inicien a la hora del arranque. Mientras menos programas se ejecuten, menos probabilidades hay que un "bug" o error de seguridad le afecte.

- **Perturbación de las Redes:** actualmente las redes se encuentran ampliamente digitalizadas y controladas por servidores, pero en el pasado la razón de perturbación de la red más frecuente era un fallo en el sistema que controla la red y los ataques a las redes estaban dirigidos principalmente a dichos servidores. En la actualidad, los ataques más peligrosos se concretan a los puntos débiles y más vulnerables de los componentes de las redes como son sistemas operativos, routers, servidores de nombres de dominio (Servidores DNS), entre otros equipos.

*Nota: Los riesgos se presentan en orden alfabético.*

### 3.4.2 ANÁLISIS DE RIESGO

Un análisis de riesgo es el resultado de responder a las siguientes preguntas:

- ¿Qué quiero proteger? Mis recursos: Personal, información, hardware, software, documentación, consumibles, entre otros.
- ¿De quién necesito protegerlo? De cualquiera que constituya una amenaza, en cualquiera de estos rubros:
- Acceso no autorizado: Utilizar recurso de cómputo sin previa autorización
- Daño a la información: Modificación o eliminación de la información en el sistema
- Robo de información: Acceso a cierta información sin previa autorización
- Divulgación de la información: Publicar detalles del sistema, como podrían ser las contraseñas, secretos industriales, investigaciones, o cualquier otro conocimiento restringido.
- Negación del servicio: Obligar al sistema a negar recursos a usuarios legítimos
- ¿Qué tantos recursos estoy dispuesto a invertir?
- ¿Cómo puedo/debo protegerlo?

Estos dos últimos puntos, dependen por completo de la situación del cliente y de los requerimientos del sistema u organización que deseamos proteger. Para nuestra aplicación responderemos estas preguntas (ver Tabla 3-3).

**Tabla 3-3 Respuesta a los cuestionamientos básicos de seguridad**

| Cuestionamiento | Respuesta |
|---|---|
| ¿Qué quiero proteger? | Servidores de producción<br>Confidencialidad de la información almacenada en la base de datos<br>Integridad de la medio ambiente en el cual se ejecuta nuestra aplicación (Recursos de la lógicos y físicos de la Secretaria del Medio Ambiente del Distrito Federal) |
| ¿De quién necesito protegerlo? | De todo aquel solicitante no registrado para tener acceso a los recursos ofrecidos por la solución Web.<br>De los mismos usuarios registrados, vigilando que sus privilegios sean adecuados con las labores que necesitan realizar en el sistema. |
| ¿Qué tantos recursos estoy dispuesto a invertir? | La Secretaria del Medio Ambiente del Distrito Federal, será la encargada de asignar los recursos que considere adecuados. |
| ¿Cómo puedo/debo protegerlo? | La aplicación cuenta con una sección programada para ofrecer un cierto nivel de seguridad, así mismo se indican las configuraciones mínimas, para lograr un nivel de seguridad.<br>Así mismo se entregan políticas y recomendaciones para reforzar la seguridad. |

### 3.4.3 POLÍTICAS Y PROCEDIMIENTOS DE SEGURIDAD

Las Políticas de seguridad son los documentos que describen, principalmente, la forma adecuada de uso de los recursos de un sistema de cómputo, las responsabilidades y derechos que tanto usuarios como administradores tienen y los procedimientos ante un incidente de seguridad. Mientras las políticas indican el "qué", los procedimientos indican el "cómo". Los procedimientos son los que nos permiten llevar a cabo las políticas.

Las políticas son parte fundamental de cualquier esquema de seguridad eficiente. En el caso de los administradores de red, les permite aminorar los riesgos, y ofrecer una respuesta más rápida y acertada en caso de presentarse un incidente. A los usuarios, les indica la manera adecuada de usar un sistema, indicando lo que puede hacerse y lo que debe evitarse en un sistema de cómputo, contribuyendo a evitar los usos no adecuados de la red sin saberlo. El tener un esquema de políticas facilita grandemente la introducción de nuevo personal, teniendo ya una base escrita y clara para capacitación; dan una imagen profesional a la organización y facilitan una auditoria.

Los principales puntos que deben contener las políticas de seguridad son:

- Ámbito de aplicación
- Análisis de riesgos
- Enunciados de políticas
- Sanciones
- Sección de uso ético de los recursos de cómputo
- Sección de procedimientos para el manejo de incidentes

Al diseñar un esquema de políticas de seguridad, conviene que dividamos nuestro trabajo en diferentes políticas específicas a un campo (v. gr. cuentas, contraseñas, control de acceso, respaldos, correo electrónico, seguridad física, personal, entre otros).

### 3.4.4 METODOLOGÍAS DE DESARROLLO

Un esquema de políticas de seguridad debe llevar ciertos pasos, para garantizar su funcionalidad y permanencia en la institución. Nuestra propuesta implica seguir los pasos que detallamos a continuación:

- Preparación – La recopilación de todo tipo de material relacionado con cuestiones de seguridad en la organización: Manuales de procedimientos, planes de contingencia, cartas compromiso, entre otros.
- Redacción – Escribir las políticas de una manera clara, concisa y estructurada. Requiere de la labor de un equipo en el que participen abogados, directivos, usuarios y administradores.
- Edición – Reproducir las políticas de manera formal para ser sometidas a revisión y aprobación
- Aprobación – Probablemente, la parte más difícil del proceso, puesto que es común que la gente afectada por las políticas se muestre renuente a aceptarlas. En esta tapa es fundamental contar con el apoyo de los directivos.
- Difusión – Dar a conocer las políticas a todo el personal de la organización mediante proyecciones de video, páginas Web, correo electrónico, cartas compromiso, entre otros.
- Revisión – Las políticas son sometidas a revisión por un comité, que discutirá los comentarios emitidos por las personas involucradas.
- Aplicación – Una política que no puede implementarse o hacerse cumplir, no tiene ninguna utilidad.
- Actualización – En el momento requerido, las políticas deberán ser revisadas y actualizadas, respondiendo a los cambios en las circunstancias. El momento ideal es justo después de que ocurra un incidente de seguridad.
- Antes de comenzar a desarrollar las políticas de seguridad, tenemos que preguntarnos:
- ¿Quién debe poder usar los recursos? – Sólo personal autorizado: Estudiantes, profesores, usuarios externos, investigadores, entre otros.
- ¿Qué constituye un uso adecuado de los recursos? – ¿Qué actividades están permitidas? ¿Qué actividades están prohibidas?
- ¿Quién debe poder proporcionar acceso al sistema? – Si no se tiene control sobre quién está dando acceso al sistema, tampoco se podrá tener control sobre quién lo utiliza.
- ¿Quién debe tener privilegios de administrador? – Debe utilizarse el principio del mínimo privilegio: Proporcionar sólo los privilegios suficientes para ejecutar las tareas necesarias
- ¿Cuáles son los derechos y responsabilidades de los usuarios? – ¿Existen restricciones en cuanto al consumo de recursos? ¿Cuáles son? ¿Qué constituye un abuso en términos de desempeño del sistema? ¿Qué requerimientos deben cumplir

las contraseñas de los usuarios? ¿Debe el usuario hacerse responsable de sus propios respaldos de información? ¿Puede el usuario divulgar información propietaria? ¿Qué tan privado es el correo electrónico de los usuarios? ¿Pueden los usuarios subscribirse a cualquier lista de discusión o grupo de noticias?

- ¿Cuáles son los derechos y responsabilidades de los administradores? – ¿Pueden monitorear o leer los archivos de los usuarios? ¿Tienen derecho a examinar el tráfico de red para un equipo en específico? ¿Tienen derecho a examinar el tráfico de toda la red? ¿A qué grado pueden hacer uso de sus privilegios? ¿Qué tanto deberán respetar la privacidad de la información de los usuarios? ¿Cómo deben resguardar su contraseña? ¿Cómo deben manejar información sensitiva?
- ¿Cómo debe manejarse la información sensitiva? – Debe evitarse que los usuarios almacenen información valiosa en sistemas poco seguros.

Es necesario tomar en cuenta diferentes escenarios, para contemplarlos a todos al crear nuestras políticas.

### 3.4.5 RECOMENDACIONES DE POLÍTICAS Y PROCEDIMIENTOS DE SEGURIDAD

En busca de lograr un nivel de seguridad aceptable, proponemos la generación de políticas y procedimientos enfocados en las siguientes actividades:

- Ingreso de un nuevo usuario al sistema
- Conexión de nuevos equipos a la red corporativa
- Localización física y lógica de los equipos de computo
- Actualizar el sistema operativo
- Control y administración del software
- Actualizaciones críticas del software
- Metodología de respaldos y restauración de información
- Programación de eventos ante un incidente de seguridad

Consideramos que el desarrollo y su posterior implementación de políticas o procedimientos dirigidos a controlar los eventos antes mencionados, traerán cambios significativos al modo en como se realizan las actividades cotidianas, pasando de acciones caóticas a un esquema estratificado de controles que permitan granular las acciones de los usuarios, implicando un sentimiento de burocracia, fenómeno que en posteriores revisiones a las políticas y procedimientos podrá atenuarse debido a la experiencia y retroalimentación que el grupo encargado de generarlas reciba de usuarios y administradores. Un ambiente con las políticas adecuadas permite mejorar la administración y supervisión de los recursos disponibles.

### 3.4.6 POLÍTICAS Y PROCEDIMIENTOS RECOMENDADOS

Con la finalidad de afectar al minino la estructura y organización de nuestro cliente, proponemos la implementación de un sistema de políticas y procedimientos básicos de fácil implementación y uso, los cuales serán empleados a nivel de la aplicación y de ser posible seguidos por todo el personal asignado por nuestro cliente para la administración de la aplicación.

#### 3.4.6.1 POLÍTICAS DE CUENTAS

Establecen qué es una cuenta de usuario para la solución Web, cómo está conformada, a quién puede serle otorgada, quién es el encargado de asignarlas, cómo deben ser creadas y comunicadas.

Esto es:

- Las cuentas deben ser otorgadas exclusivamente a usuarios legítimos y previamente autorizados por la Secretaría del Medio Ambiente del Distrito Federal.
- Una cuenta deberá estar conformada por un nombre de usuario y contraseña, identificación completa del usuario, así como el privilegio de acceso que se le concede dentro de la aplicación.
- El nombre de usuario de una cuenta deberá estar conformado por la primera letra de su nombre seguido de su apellido paterno y la ultima letra de su apellido materno

#### 3.4.6.2 POLÍTICAS DE CONTRASEÑAS

Son una de las políticas más importantes, ya que por lo general, las contraseñas constituyen la primera y tal vez única manera de autenticación y, por tanto, la única línea de defensa contra ataques. Éstas establecen quién asignará la contraseña, qué longitud debe tener, a qué formato deberá apegarse, cómo será comunicada, entre otros aspectos.

Esto es:

- La longitud de una contraseña deberá siempre ser verificada de manera automática al ser construida por el personal encargado de generar cuentas. Todas las contraseñas deberán contar con al menos ocho caracteres.
- Todas las contraseñas elegidas por los usuarios deben contar con un nivel adecuado de dificultad para ser descifradas. No deben ser utilizadas palabras que aparezcan en el diccionario, secuencias conocidas de caracteres, datos personales ni acrónimos.
- Está prohibido que los usuarios construyan contraseñas compuestas de algunos caracteres constantes y otros que cambien de manera predecible y sean fáciles de descifrar
- Los usuarios no deben construir contraseñas idénticas o muy parecidas a contraseñas anteriores.

### 3.4.6.3 POLÍTICAS DE CONTROL DE ACCESO

Especifican cómo deben los usuarios acceder al sistema, desde dónde y de qué manera deben autentificarse.

Esto es:

- Todos los administradores del sistema deberán acceder al sistema desde los equipos asignados a la administración de los servidores y aplicaciones que dependan de ellos.
- Está prohibido acceder al sistema con una cuenta diferente de la propia, aún con la autorización del dueño de dicha cuenta
- Si un usuario está fuera del sitio de trabajo, deberá vigilar que el equipo que use cuente con un antivirus, no dejar datos personales almacenados de alguna forma en el equipo.
- Al momento de ingresar al sistema, cada usuario deberá ser notificado de la fecha, hora y dirección desde la que se conectó al sistema por última vez, lo cual permitirá detectar fácilmente el uso no autorizado del sistema
- Está terminantemente prohibido ejecutar programas que intenten descifrar las contraseñas alojadas en los sistemas de usuarios de máquinas locales o remotas
- La cuenta de un usuario es personal e intransferible, por lo cual no se permite que este comparta su cuenta ni su contraseña con persona alguna, aún si ésta acredita la confianza del usuario
- Está estrictamente prohibido hacer uso de herramientas tales como programas que rastrean vulnerabilidades en sistemas de cómputo propios o ajenos
- Está estrictamente prohibido hacer uso de programas que explotan alguna vulnerabilidad de un sistema para proporcionar privilegios no otorgados explícitamente por el administrador
- No se permite bajo ninguna circunstancia el uso de cualquiera de las computadoras con propósitos de ocio o lucro

### 3.4.6.4 POLÍTICAS DE RESPALDOS

Especifican qué información debe respaldarse, con qué periodicidad, qué medios de respaldo utilizar, cómo deberá ser restaurada la información, dónde deberán almacenarse los respaldos, entre otros temas.

Esto es:

- El administrador del sistema es el responsable de realizar respaldos de la información periódicamente Cada treinta días como mínimo deberá efectuarse un respaldo completo del sistema y cada día deberán ser respaldados todos los archivos críticos para los servidores de producción más importantes. Durante el periodo de mayor actividad de la solución Web propuesta, se recomienda un respaldo por día de la base de datos principal de la aplicación.
- La información respaldada deberá ser almacenada en un lugar seguro y distante del sitio de trabajo

- Deberá mantenerse siempre una versión impresa de los archivos de licencias del software que lo requiera.
- En el momento en que la información respaldada deje de ser útil a la organización, dicha información deberá ser borrada antes de deshacerse o reutilizar los medios.

### 3.4.6.5 POLÍTICAS DE AUDITORÍA DEL SISTEMA

Establecen los lineamientos bajo los cuales pueden ser monitoreadas las actividades de los usuarios del sistema de cómputo, así como la manera en que debe manejarse la auditoría del sistema y el propósito de la misma.

Esto es:

- Deberán ser registrados en bitácoras todos los accesos a la base de datos emitidos por todos los usuarios del sistema, para propósitos de auditoría.
- Cada semana deberá hacerse el un respaldo de la información producto de la auditoría, cifrándose y respaldándose la información en un dispositivo de almacenamiento permanente

### 3.4.6.6 VIOLACION A LAS POLÍTICAS DE SEGURIDAD

Tarde o temprano, todas las políticas serán violadas. Los motivos pueden ser varios, entre los más típicos que se presentan y que requieren de una atención inmediata son:

- Intrusiones externas
- Indicios de sabotaje interno
- Negligencia
- Error accidental
- Desconocimiento de la misma
- Falta de entendimiento de la misma

Cada una de ellas requiere de un seguimiento personalizado para prevenir nuevos incidentes. La gravedad de las mismas requiere de sanciones adecuadas, debido a que una violación a las políticas de seguridad debida al desconocimiento o falta de entendimiento de las mismas, se origina por una inadecuada propagación o correcta redacción de las mismas y es deber del departamento encargado de la generación de las políticas revisar este punto. En el otro extremo encontramos las violaciones causadas por intrusos externos o sabotaje interno, lo cual requiere de la acción de los departamentos legales para deslindar las responsabilidades y evaluar los daños causados. Para todos los casos las acciones a tomar una vez ocurrido el incidente son:

- Investigar quién llevó a cabo esta violación
- Investigar cómo y por qué ocurrió esta violación
- Aplicar una acción correctiva (disciplinaria)

La valoración de los daños causados por una violación a las políticas de seguridad, pueden llevar a darnos cuenta que un usuario de la institución ha causado una violación de las

políticas de un sitio remoto (otra institución o persona), motivo por el cual se deberán considerar las siguientes acciones:

- Debe haber acciones a seguir bien definidas con respecto a los usuarios corporativos
- Debe contarse con una protección adecuada en contra de posibles acciones provenientes del sitio remoto

Es así que debemos saber que curso de acción tomar ante un incidente de seguridad, para ello podemos emplear dos estrategias básicas:

### 3.4.6.7 PROTEGER Y PERSEGUIR

- Su principal objetivo es proteger y preservar los servicios del sitio, y restablecerlos lo más rápido posible
- Se realizan acciones drásticas, tales como dar de baja los servicios, desconectar el sistema de la red, suspender funciones, entre otras acciones.
- Estrategia que se emplea si:
  - Los activos están bien protegidos
  - Se corre un gran riesgo debido a la intrusión
  - No existe la posibilidad o disposición para enjuiciar
  - Se desconocen las intenciones del intruso
  - Los usuarios son poco sofisticados y su trabajo es vulnerable
  - Los recursos de los usuarios son minados.

### 3.4.6.8 PERSEGUIR Y ENJUICIAR

- Su objetivo principal es permitir que los intrusos continúen con sus actividades en el sistema hasta que pueda identificarse a los responsables.
- Estrategia que se emplea si:
  - Los recursos están bien protegidos
  - Se dispone de respaldos confiables
  - El riesgo para los activos es mayor que el daño de esta y futuras intrusiones
  - El ataque proviene de un instituciones con las que guardamos cierta relación, y ocurre con cierta frecuencia e intensidad
  - El sistema posee cierta atracción para los intrusos
  - El sistema está dispuesto a correr el riesgo a que se exponen los activos al permitir que el ataque continúe
  - Puede controlarse el acceso al intruso
  - Se cuenta con herramientas de seguridad confiables
  - El personal técnico conoce a profundidad el sistema operativo y sus utilerías
  - Existe disposición para la persecución por parte de los directivos
  - Existen leyes al respecto (Motivo por el cual en México no es muy empleado)

Para el desarrollo de las políticas existen dos enfoques: permisivo (todo lo que no esté explícitamente prohibido está permitido) y restrictivo (todo lo que no esté explícitamente permitido está prohibido). La selección de uno u otro enfoque dependerá del tipo de organización y el nivel de seguridad que esta requiera. Para fines convencionales se

propone el empleo de un enfoque permisivo. El empleo de este enfoque conducirá a detectar las acciones que requieren de supervisión y control, con el paso del tiempo se puede llegar a implementar políticas usando un enfoque restrictivo con la certeza de encontrar pocos incidentes que no hayan sido previstos con anterioridad.

## 3.5 SEGURIDAD FÍSICA

El primer pasó a considerar en un esquema de seguridad, que muchas veces no recibe suficiente atención, es la seguridad física – las medidas que se usan para proteger las instalaciones en las que reside un sistema de cómputo: llaves, candados, tarjetas de acceso, puertas, ventanas, alarmas, vigilancia. Las recomendaciones respecto a la seguridad física incluyen:

- Mantener las computadoras alejadas del fuego, humo, polvo y temperaturas extremas.
- Colocarlas fuera del alcance de rayos, vibraciones, insectos, ruido eléctrico (balastras, equipo industrial, o cualquier otra fuente que se considere de riesgo.), agua o líquidos de cualquier naturaleza.
- Mantener las computadoras alejadas de comida y bebida.
- No desatender las sesiones de trabajo activas.

Dejamos fuera de los alcance de las políticas de seguridad propuestas a objetos como la seguridad en el medio de comunicación, la seguridad en la estructura de dominios y la seguridad en recursos remotos, esto debido a que son entidades que requieren conocimientos exactos de la estructura disponible en las instalaciones de nuestro cliente y pueden requerir cambios de fondo a la arquitectura de la organización.

## 3.6 SEGURIDAD A NIVEL APLICACIÓN

Los esfuerzos realizados en obtener una aplicación estable y siempre disponible, requieren el trabajo de administradores de red, administradores de servidores y los esfuerzos constantes de un grupo dedicado al soporte de los usuarios finales. En el caso de los arquitectos y desarrolladores, sus labores no deben de terminar en el desarrollo de interfaces y componentes que ayuden a los usuarios finales a obtener el valor agregado que esperan de nuestra solución. La necesidad de contar con un ambiente seguro para la ejecución de nuestra aplicación requiere del trabajo de los administradores y del personal de red, el cual se integrara con una planeación, desarrollo e implementación de esquemas de seguridad por parte de los arquitectos y desarrolladores involucrados en el proyecto. Es así que: las aplicaciones que almacenan información confidencial deben adoptar medidas de protección frente a ataques malintencionados y competidores que intenten apropiarse de información o propiedad intelectual. A la hora de diseñar un modelo de seguridad para una aplicación, es necesario considerar los requisitos de seguridad desde una perspectiva empresarial y las implicaciones que el modelo seleccionado pueda tener en el rendimiento, la escalabilidad y la distribución. Siempre con el apoyo de los niveles gerenciales, los cuales facilitaran la propagación y adopción de los esquemas de seguridad propuestos.

### 3.6.1 APLICACIONES BASADAS EN WEB

En el capitulo 2 y a lo largo del presente, se describe la naturaleza de nuestra aplicación y sus diversos componentes, motivo por el cual nos enfocaremos en el escenario de las aplicaciones basadas en Web que son distribuidas a través de Internet. Las aplicaciones de Internet cuentan con un público muy extenso, ofrecen una gran variedad de usos posibles y exigen una serie de requisitos de seguridad. Pueden presentare en forma de aplicaciones de portal que no requieren autenticación de usuario o aplicaciones Web que ofrecen contenido para usuarios registrados, Ambos caso son de nuestro interés, al contar con una aplicación que será accedida por un grupo reducido, los cuales contarán con contenido dinámico, a su vez se cuentan con módulos de libre acceso pero carentes de un contenido privado.

Al desarrollar aplicaciones basadas en Web se buscan los mecanismos de defensa adecuados y con la posibilidad de una sencilla escalabilidad. Algunos puntos que se requieren revisar son:

- Selección de un almacén de credenciales de usuario adecuado (e.i. una base de datos personalizada o un servicio de directorio "Active Directory".)
- Conseguir que la aplicación funcione a través de los servicios de seguridad existentes.
- Transferir las credenciales de seguridad a través de los distintos niveles de la aplicación.
- Llevar a cabo la autorización de los clientes.
- Garantizar la integridad y privacidad de los datos conforme se distribuyen a través de redes públicas e internas.
- Garantizar la seguridad del estado de la aplicación con una base de datos.
- Asegurar la integridad de los datos de la aplicación.
- Implementar una solución que pueda escalarse para admitir un mayor número de usuarios.

Antes de proceder a la descripción del modelo de seguridad implementado en esta aplicación, es necesario describir los métodos involucrados en la conformación de este modelo, algunos métodos pueden estar solo disponibles en la plataforma Microsoft Windows que empleada como soporte de nuestra aplicación, algunos otros serán métodos comunes a otras plataformas. De igual forma, algunas funciones son exclusivas del lenguaje de programación seleccionado, en los casos donde es posible, ambos elementos (métodos y funciones) han sido tratadas para mantener una lectura ágil de la presente sección.

### 3.6.2 RELACIÓN ENTRE "IIS" Y ASP .NET

A la hora de diseñar una aplicación es necesario comprender la relación existente entre la autenticación de los servicios utilizados por el servidor Web y el lenguaje de programación utilizado. En nuestro escenario se utilizaran los servicios de Internet Information Services ("IIS" 5.0) y la arquitectura de seguridad de Microsoft ASP .NET. De este modo, se podrá autenticar a los usuarios de forma adecuada y obtener el contexto de seguridad correcto en la aplicación. Se debe tener en cuenta que la configuración de seguridad de ASP .NET y la

de "IIS" son totalmente independientes y que se pueden utilizar por separado o de forma conjunta.

Para el caso de "IIS", los valores de configuración relacionados con la seguridad se mantienen en la metabase "IIS". Para el caso de ASP .NET, esta información se mantiene en los archivos de configuración XML. Mientras que esta operación simplifica en general la distribución de la aplicación desde el punto de vista de la seguridad, el modelo de seguridad adoptado por la aplicación necesitará la configuración correcta tanto de la metabase "IIS" como de la aplicación ASP .NET a través de su archivo de configuración. En la Figura 3-5 se muestra la relación existente entre "IIS" y ASP .NET

La Figura 3-5 ejemplifica el proceso de validación de una solicitud realizada por un cliente a una aplicación Web desarrollada con ASP.NET, este proceso puede variar en función de los requerimientos de nuestra aplicación, motivo por el cual podemos realizar una autenticación de forma óptima en función de nuestras necesidades. Algunos elementos presentados como son la "representación", la comprobación de permisos "NTFS", serán comentados en secciones posteriores de este capitulo.

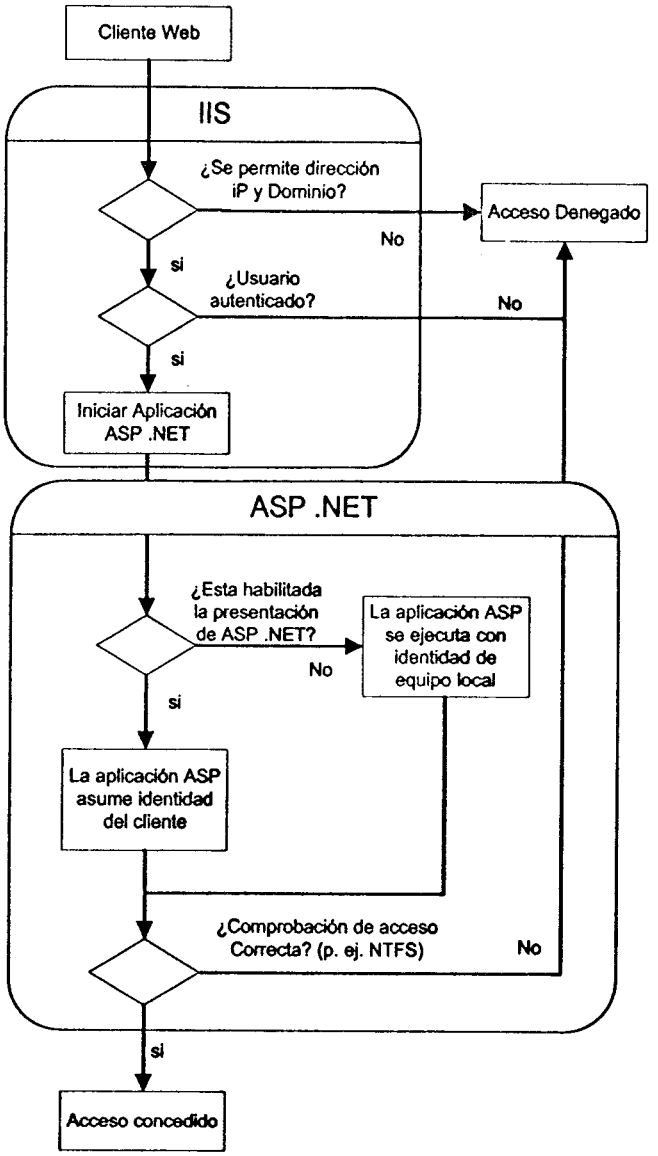



**Figura 3-3 Relación existente entre "IIS" y ASP .NET.**

### 3.6.3 PROVEEDORES DE AUTENTICACIÓN DE ASP .NET Y SEGURIDAD DE "IIS"

Cuando hablamos de la necesidad de reforzar la seguridad de nuestro proyecto, recurrimos a técnicas de seguridad que se integran con el desarrollo del sistema. Una de las primeras prioridades es solicitar y procesar la identidad de cliente que desee hacer uso de esta aplicación (proceso que es mostrado en la Figura 3-3), motivo por el cual se recurre a los proveedores de autenticación. Para el caso de ASP.NET la implementación del proceso de autenticación, utiliza proveedores para la misma; se trata de módulos de código que

comprueban las credenciales e implementan otras funcionalidades de seguridad. En la Tabla 3-4 se presentan los proveedores de autenticación utilizados por ASP.NET que pueden ser utilizados en un proceso de validación para "IIS".

**Tabla 3-4 Proveedores utilizados por ASP.NET**

| PROVEEDOR | DESCRIPCIÓN |
|---|---|
| Autenticación de Windows | Este proveedor hace uso de las capacidades de autenticación de "IIS". Una vez que "IIS" completa su autenticación, ASP .NET utiliza el vale de la identidad autenticada para autorizar el acceso. |
| Autenticación por formularios | El uso de este proveedor permite que las solicitudes sin autenticación se redirijan a un formulario HTML especificado utilizando el redireccionamiento del lado del cliente. A continuación, el usuario podrá proporcionar las credenciales de inicio de sesión y volver a enviar el formulario al servidor. Si la aplicación autentica la solicitud (empleando la lógica específica de la aplicación), ASP .NET envía un cookie que contenga las credenciales o una clave para volver a adquirir la identidad del cliente. Las solicitudes posteriores se envían con el cookie en el encabezado de las mismas, lo que significa que son innecesarias futuras autenticaciones. |
| Autenticación de Passport | Se trata de un servicio de autenticación centralizado que proporciona Microsoft y que ofrece un medio único de inicio de sesión, así como servicios de suscripción para los sitios de participación. ASP .NET, junto al kit de desarrollo de software (SDK) de Microsoft® Passport, proporciona una funcionalidad similar a los usuarios de Passport y de autenticación por formularios. |

### 3.6.4 AUTENTICACIÓN SOPORTADA POR ASP.NET

Una vez realizado el llamado de los proveedores de autenticación, ASP.NET proporciona un mecanismo de representación para establecer el vale de seguridad del subproceso de la aplicación. La obtención del vale correcto depende de la configuración adecuada de la autenticación de "IIS", de los proveedores de autenticación de ASP .NET y de los valores de la representación de ASP.NET. En la Figura 3-4 se muestran las combinaciones posibles que se pueden producir entre la autenticación de "IIS" y los proveedores de autenticación de ASP .NET.

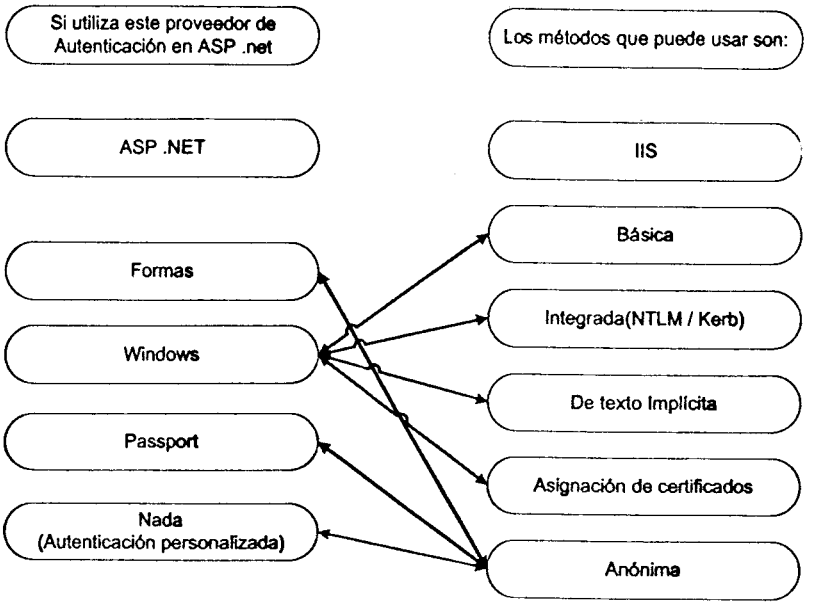



**Figura 3-4 Relación entre los proveedores de autenticación de ASP.net y los procesos de validación en "IIS"**

El trabajo realizado en conjunto por IIS y ASP.NET permite adecuar las necesidades de seguridad al entorno en el cual nos desempeñamos, de tal forma que es posible presentar un esquema de autenticación que refleje los intereses de nuestro cliente.

### 3.6.5 AUTENTICACIÓN CON CUENTAS DE WINDOWS

Si se planea autenticar a los usuarios utilizando las cuentas mantenidas por un controlador de dominios, se debe utilizar la autenticación de "IIS" en combinación con el proveedor de Windows para ASP.NET, como se muestra en la Figura 3-4. Con este enfoque, no es necesario escribir ningún código de autenticación específico.

#### 3.6.5.1 AUTENTICACIÓN CON CUENTAS QUE NO PERTENECEN AUN DOMINIO

Si se planea autenticar a los usuarios a nivel de la aplicación y éstos no disponen de cuentas de locales o de un dominio, lo recomendado será configurar "IIS" para que utilice la autenticación anónima. En este tipo de configuración se deben tener en cuenta los módulos de autenticación presentados en la Tabla 3-5.

**Tabla 3-5 Proveedores de autenticación para usarse con cuentas que no pertenecen a un dominio**

| Proveedor | Descripción |
|---|---|
| None | Se utiliza cuando no se está autenticando a ningún usuario, o bien, si se está desarrollando código de autenticación personalizado. |
| Forms | Se emplea cuando se desea proporcionar a los usuarios una página de inicio de sesión. |
| Passport | Se utiliza cuando se hace uso de los servicios Passport. |

No existe una combinación entre proveedores y métodos de autentificación que garantice un mejor rendimiento o un nivel mas alto de seguridad, cada combinación cubrirá las necesidad de autentificación que se tengan, es deber del arquitecto y del grupo directivo optar por la combinación que mejor refleje los requerimientos de la aplicación y la ideología corporativa que se tenga. En las secciones 3.6.5, 3.6.5.1 se presentaron los modos de autenticación recomendados para interactuar con los sistemas "Windows".

### 3.6.5.2 MODELOS DE AUTENTIFICACIÓN Y SUS ESCENARIOS DE USO

Los métodos de autentificación presentados en la Figura 3-4, constituyen un conjunto de mecanismos adecuados para lograr los objetivos de validación, cada método presenta características que ofrecen un mejor rendimiento en función del escenario en el cual se pretendan utilizar, a continuación describiremos brevemente la función de cada método. La implementación de cada uno puede ser consultada en el artículo "Crear aplicaciones ASP.NET seguras", que se puede obtener del sitio de Microsoft Latinoamérica.

### 3.6.5.3 AUTENTICACIÓN ANÓNIMA

Con este tipo de autenticación, el servidor no solicita al cliente que envíe las credenciales de usuario. Constituye una buena opción cuando el sitio o el servicio se encuentran disponibles públicamente y no es necesario conocer la identidad del autor de la llamada. Asimismo, no suelen existir restricciones del explorador que procedan de incompatibilidades con los mecanismos de autenticación admitidos. Cuando se configura un sitio para la autenticación anónima, se permite el acceso a todos los usuarios. Es importante tener en cuenta que, aunque "IIS" se puede haber configurado para este tipo de autenticación, ésta se puede estar realizando en la capa de ASP.NET, lo que no constituye una autenticación anónima en toda regla. En esta sección se asume que tanto "IIS" como la aplicación no requieren el inicio de sesión.

#### 3.6.5.3.1 ESCENARIOS TÍPICOS DE USO

Se debe considerar el uso de la autenticación anónima cuando:

- No es necesario conocer el nombre y/o la contraseña del autor de la llamada para el inicio de sesión o los componentes de lógica empresarial.
- La información que se está protegiendo se considera "pública".
- No se debe considerar el uso de la autenticación anónima cuando:

- La base de usuarios se limita al requerir un nombre de inicio de sesión y una contraseña.

### 3.6.5.4 AUTENTICACIÓN BÁSICA

Cuando "IIS" se configura para la autenticación básica, indica al explorador que envíe las credenciales del usuario a través del protocolo HTTP. Las contraseñas y los nombres de usuario se codifican utilizando el algoritmo de codificación Base64. Aunque la contraseña está codificada, se considera insegura debido a que se puede descifrar con relativa facilidad. El explorador muestra al usuario un cuadro de diálogo y, a continuación, vuelve a enviar la solicitud anónima original con las credenciales suministradas, incluyendo el nombre de usuario y la contraseña. Un cuadro de diálogo de inicio de sesión emergente puede ser o no adecuado dependiendo de los requisitos de diseño de la interfaz de usuario. La mayor parte de los exploradores de Internet admiten la autenticación básica.

#### 3.6.5.4.1 ESCENARIOS TÍPICOS DE USO

- Se debe considerar el uso de la autenticación básica cuando:
- Los usuarios disponen de cuentas de Dominio de Windows NT o Active Directory.
- Es necesario admitir varios tipos de exploradores, incluidos Netscape Navigator y todas las versiones de Internet Explorer (incluyendo Pocket PC y las plataformas de Windows CE).
- Es necesario admitir la autenticación en Internet.
- Es preciso tener acceso a la contraseña sin cifrar en el código de la aplicación.
- Se debe admitir la delegación.

### 3.6.5.5 AUTENTICACIÓN DE TEXTO IMPLÍCITA

Este tipo de autenticación resulta novedosa para Windows 2000 y "IIS" 5.0. Con la autenticación de texto implícita se cifra la información de la contraseña del usuario y se proporciona un mecanismo que ayuda a evitar algunos ataques comunes del servidor (por ejemplo, un ataque de reproducción). Con esta autenticación las credenciales no se envían a través de la red utilizando el texto sin cifrar, tal y como sucede con la autenticación básica, sino que se emplea un mecanismo de hash denominado MD5 desarrollado por RSA. (Para obtener información más detallada, consulte el texto (en inglés) "The MD5 Message-Digest Algorithm" en http://www.ietf.org/rfc/rfc1321.txt.). Aunque se trata de una opción de autenticación viable para los escenarios de Internet, los requisitos del cliente y del servidor limitan su uso generalizado. A diferencia de la autenticación básica y de modo similar a NTLM y Kerberos, "IIS" conecta al usuario localmente en el servidor Web, por lo que no se puede realizar la delegación.

#### 3.6.5.5.1 ESCENARIOS TÍPICOS DE USO

Se debe considerar el uso de la autenticación de texto implícita cuando:

- El servidor Web ejecuta Windows 2000 y los usuarios disponen de cuentas de Windows almacenadas en Active Directory.
- Todos los clientes utilizan la plataforma .NET o Internet Explorer 5.x.

- Es necesario disponer de un mayor nivel de cifrado de contraseña que el que proporciona la autenticación básica.
- Es preciso admitir la autenticación en Internet.

### 3.6.5.6 AUTENTICACIÓN DE WINDOWS INTEGRADA

Este tipo de autenticación (que utiliza desafío/respuesta de NTLM o Kerberos) implica la autenticación del usuario con un Dominio de Windows NT o con cuentas de Active Directory. Al contrario de lo que sucede con la autenticación básica y de texto implícita, la contraseña cifrada no se envía a través de la red, lo que hace que este método resulte bastante seguro. Si se instalan los servicios de Active Directory en el servidor y el explorador es compatible con el protocolo de autenticación Kerberos V5, se utilizará dicho protocolo así como el de desafío/respuesta; de lo contrario, sólo se utilizará este último. Este protocolo es el más adecuado para un entorno de Intranet, donde tanto el usuario como los equipos del servidor Web se encuentran en el mismo dominio y los administradores pueden asegurarse de que cada equipo está ejecutando la versión 3.01 o posterior de Microsoft Internet Explorer.

#### 3.6.5.6.1 ESCENARIOS TÍPICOS DE USO

Se debe considerar el uso de la autenticación de Windows integrada cuando:

- Los usuarios disponen de cuentas de Dominio de Windows NT o Active Directory.
- La aplicación se ejecuta en una intranet (con un servidor de seguridad).
- Todos los clientes ejecutan la versión 3.01 o posterior de Internet Explorer.
- Es necesario ejecutar la delegación (para lo cual se requiere Kerberos).
- Es preciso disponer de un procedimiento de inicio de sesión sencillo para los usuarios del dominio (por ejemplo, sin cuadros de diálogo emergentes de inicio de sesión).

### 3.6.5.7 AUTENTICACIÓN DE CERTIFICADOS

Un certificado es una "clave" digital que se encuentra instalada en el equipo. Cuando el equipo intenta conectarse al servidor, se presentará automáticamente la clave con el fin de autenticar al usuario. Los certificados de cliente se pueden asignar a las cuentas de Windows en un dominio o en Active Directory. Si se utiliza Windows Authentication Provider en ASP .NET, el subproceso de la aplicación se ejecutará como el usuario al que se le asigna el certificado. Asimismo, se puede implementar la autenticación personalizada en ASP .NET de modo que, por ejemplo, se pueda utilizar la dirección de correo electrónico (o igualmente un campo único) incluida en el certificado. Desde la perspectiva del cliente, la seguridad no presenta problemas, ya que no es necesario que el cliente inicie la sesión utilizando una página destinada para ello. Esto hace que los certificados constituyan una opción atractiva para los procesos empresariales automatizados.

### 3.6.5.8 ESCENARIOS TÍPICOS DE USO

Se debe considerar el uso de la autenticación de certificados cuando:

- Los datos que se están protegiendo poseen un alto grado de confidencialidad y es necesario disponer de una solución muy segura.
- Se requiere la autenticación mutua.
- Se desea que un tercero pueda administrar la relación entre el servidor y el titular del certificado.
- Se desea que la interacción del cliente no presente problemas; por ejemplo, en un intercambio automatizado entre empresas.

### 3.6.5.9 AUTENTICACIÓN DE PASSPORT

La autenticación de Passport es un servicio de autenticación centralizado que proporciona Microsoft. Cuando se utiliza, en algunos casos no es necesario implementar un código de autenticación, una página de inicio ni una tabla de usuarios propios. Passport funciona empleando un mecanismo de "cookies". Si los clientes se han autenticado en Passport con anterioridad, podrán tener acceso al sitio. De lo contrario, se les redirigirá de forma automática al sitio de Passport para realizar la autenticación.

Este método constituye una buena opción si se requiere una capacidad de inicio de sesión única a través de varios dominios que también admitan Passport. Este mecanismo proporciona servicios adicionales además de su función como servicio de autenticación, incluyendo la administración de perfiles y los servicios de compra.

En la plataforma de Windows 2000, no existe una integración directa de Passport en ningún mecanismo de autenticación y autorización creados en un sistema operativo. Mientras que .NET Framework sí comprueba los "cookies" de Passport, si se mantiene una base de datos de usuarios propia se debe implementar un código propio para asignar el usuario de Passport al mismo usuario, además de implementar un mecanismo de autorización personal.

#### 3.6.5.9.1 ESCENARIOS TÍPICOS DE USO

Se debe tener en cuenta el uso de la autenticación de Passport cuando:

- El sitio se va a utilizar junto con otros sitios habilitados para Passport y se desea ofrecer capacidad de un inicio de sesión único a los usuarios que tienen acceso.
- No se desea mantener una base de datos de nombres de usuario y contraseñas.

### 3.6.5.10 AUTENTICACIÓN POR FORMULARIOS

La autenticación por formularios hace referencia a un componente de interfaz de usuario personalizado que acepta las credenciales del mismo; por ejemplo, un nombre de usuario y una contraseña. Un gran número de aplicaciones de Internet utilizadas actualmente presentan este tipo de formularios para que los usuarios inicien la sesión. Es importante tener en cuenta que el formulario no realiza la autenticación por sí mismo, sino que sólo se ofrece como un modo de obtener las credenciales de usuario. La autenticación se realiza cuando se obtiene acceso al nombre y a la contraseña de usuario empleando código personalizado.

Cuando el usuario se autentica, el servidor suele ofrecer al cliente varios medios para indicar que ya ha sido autenticado para las siguientes solicitudes. Si es necesario, se puede obligar al cliente a autenticarse en cada solicitud, aunque ello podría afectar al rendimiento y a la escalabilidad. Para identificar a un cliente que ya ha iniciado la sesión con anterioridad existen dos enfoques básicos que se deben tener en cuenta:

- "cookies". Una "cookie" es una pequeña porción de datos que el servidor presenta inicialmente al cliente. Posteriormente, el cliente vuelve a mostrarla al servidor con cada solicitud HTTP. Se puede utilizar como una forma de indicar que el cliente ya se ha autenticado. ASP .NET proporciona una mecanismo para que se utilicen "cookies" en la autenticación de formularios en el módulo CookieAuthenticationProvider. Dichos "cookies" son compatibles con la mayor parte de los exploradores Web, incluidos Internet Explorer y Netscape Navigator.
- Personalización. Se puede implementar un mecanismo personalizado propio para identificar el cliente en el servidor. Si los clientes tienen deshabilitada la función de "cookies", se debe considerar el almacenamiento de un identificador único en cada cadena de consulta URL. Asimismo, se pueden utilizar campos de formulario ocultos, almacenados en un nivel superior permanente o marco no visible. En cualquier caso, es necesario asegurarse de que ningún intruso pueda simular la autenticación en la aplicación mediante programación.

### 3.6.5.10.1 ESCENARIOS TÍPICOS DE USO

Se debe considerar el uso de la autenticación de formularios cuando:

- Los nombres de usuario y contraseñas se encuentran almacenados en otras ubicaciones distintas de las cuentas de Windows. Se debe tener en cuenta que la autenticación de formularios no se puede utilizar con cuentas de Windows.
- Se está distribuyendo la aplicación a través de Internet.
- Es necesario admitir todos los sistemas operativos de los exploradores y del cliente.
- Se desea proporcionar un formulario de interfaz de usuario propio como página de inicio de sesión.

### 3.6.5.11 REPRESENTACIÓN Y DELEGACIÓN

Una vez cumplido con el proceso de Autentificación la aplicación cuenta con la posibilidad de establecer una relación entre el solicitante y los recursos existentes en el sistema (v.gr. Bases de datos, archivos, programas, entre otros), para realizar esta tarea, se cuenta con los mecanismo de "Representación" y "Delegación", los cuales fueron diseñados para transportar la identidad del cliente y presentarla ante los recursos remotos.

Con la representación, las aplicaciones ASP.NET se pueden ejecutar de forma opcional con la identidad del cliente en cuyo nombre dichas aplicaciones están funcionando. Este proceso se suele ejecutar para controlar el acceso a los recursos. Se debe considerar cuidadosamente si el proceso de representación es o no necesario, ya que el mismo supone un consumo de recursos adicionales en el servidor de la aplicación. La delegación es una

forma más eficaz de representación que permite que se tenga acceso a los recursos remotos mediante el proceso del servidor mientras se actúa como cliente.

Si se habilita el proceso de representación, ASP.NET recibirá el vale para realizar dicha representación desde "IIS". En una aplicación Web se dispone de un mayor control granular de representación cuando se utiliza ASP.NET en comparación con las páginas Active Server (ASP) tradicionales.

Si la aplicación reside en un recurso compartido UNC, ASP.NET siempre representará al vale de UNC de "IIS" para tener acceso a dicho recurso, salvo que se emplee una cuenta configurada. Si se proporciona una cuenta configurada de forma explícita, ASP.NET preferirá utilizarla antes que emplear el vale de UNC de "IIS".

En la Tabla 3-6 se muestra el vale de subprocesos que ASP .NET utiliza para ejecutar la solicitud basada en tres valores de configuración distintos de Web.config. Tenga en cuenta que la cuenta IUSR_SERVER indica la cuenta configurada para el acceso anónimo de la dirección URL actual (es decir, no tiene que ser una cuenta IUSR). La cuenta del proceso es la que el proceso de trabajo de la aplicación está ejecutando, que de forma predeterminada es la Cuenta del Sistema, a no ser que se configure de forma específica.

**Tabla 3-6 Vale de subprocesos de ASP para las configuraciones de ASP y "IIS"**

| Representación de ASP .NET | Vale usado por "IIS" | | |
|---|---|---|---|
| | "IIS" utiliza la autenticación anónima | "IIS" no utiliza la autenticación anónima | La aplicación reside en un recurso compartido UNC |
| Deshabilitada | Cuenta del proceso | Cuenta del proceso | Vale de UNC de "IIS" |
| Habilitada | IUSR_SERVER | Usuario autenticado | Vale de UNC de "IIS" |
| Habilitada con un usuario especificado "USER" | "USER" | "USER" | "USER" |

### 3.6.6 IDENTIDADES DE LA APLICACIÓN

Debido a las características de la plataforma .NET de Microsoft, toda aplicación Web, se ejecuta con una cuenta de sistema, esta cuenta es "aspnet_wp.exe", la cual puede ser otra en función del mecanismo de Representación empleado o de la configuración seleccionada en el servidor de aplicación. Se recomienda ejecutar el proceso de trabajo de la aplicación ASP.NET (aspnet_wp.exe), la cual desde la versión 1.0 del "FrameWork.NET", es una cuenta configurada con mínimos privilegios. Esta recomendación se debe a dos razones fundamentales. En primer lugar, si la seguridad se encuentra comprometida, el intruso no dispondrá de acceso de nivel administrativo. En segundo lugar, permite que los Proveedores de Servicios de Aplicaciones (ASP) puedan ejecutar aplicaciones utilizando una cuenta específica, de modo que las aplicaciones alojadas no podrán comprometer la integridad del equipo servidor ni realizar acciones que requieran privilegios administrativos.

Para el caso de que se deseé emplear una cuenta personalizada, ésta debe disponer de los permisos de acceso necesarios en los siguientes directorios.

- El acceso de lectura-escritura es necesario en el directorio de archivos temporales %installroot%\ASP .NET Temporary Files. Los subdirectorios que se encuentran bajo esta raíz se utilizan para la salida compilada dinámicamente.
- El acceso de lectura-escritura es necesario en el directorio %temp%. Los compiladores lo utilizan durante la compilación dinámica.
- El acceso de lectura es necesario en el directorio de la aplicación.
- El acceso de lectura es necesario en la jerarquía %installroot% para permitir el acceso a los ensambladores del sistema.

### 3.6.7 MÉTODOS DE AUTENTICACIÓN

Existe una variedad de opciones para realizar la autenticación en las aplicaciones Web de .NET. Por ejemplo, se puede optar por hacer uso de uno de los mecanismos de autenticación admitidos de "IIS", o bien, se puede decidir realizar la autenticación en el código de la aplicación. A la hora de seleccionar un método de autenticación se deben considerar los siguientes factores de forma independiente o en su totalidad:

- Sistemas operativos del cliente y del servidor.
- Tipo de explorador cliente.
- Número de usuarios, así como ubicación y tipo de nombre de usuario y base de datos de contraseñas.
- Consideraciones sobre la distribución como, por ejemplo, si la aplicación se encuentra basada en Internet o Intranet y si se ubica en un servidor de seguridad.
- Tipo de aplicación; por ejemplo, si se trata de un sitio Web interactivo o de un servicio Web no interactivo.
- Confidencialidad de los datos que se están protegiendo.
- Factores de rendimiento y escalabilidad.
- Requisitos de autorización de la aplicación; por ejemplo, se puede desear que la aplicación se encuentre disponible para todos los usuarios, o bien, restringir ciertas áreas de la misma a usuarios registrados o reservar otras "sólo para administradores".

### 3.6.8 DETERMINACIÓN DE UN MÉTODO DE AUTENTICACIÓN

El diagrama de flujo de la Figura 3-5 puede servir de ayuda para determinar el método de autenticación más adecuado, según los requisitos de la aplicación concreta. Para utilizar este diagrama, se deben responder a las siguientes preguntas relativas a la naturaleza de la base de usuarios y al modelo de distribución. En los extremos del diagrama se incluyen los posibles métodos de autenticación que resultan más adecuadas.

#### 3.6.8.1 EXPLICACIÓN DE LOS PUNTOS DE DECISIÓN DEL DIAGRAMA DE FLUJO

- ¿Deben los usuarios iniciar la sesión? ¿Es necesario proporcionar el nombre de usuario y la contraseña para tener acceso al sitio o al servicio?

- ¿Es necesaria la representación? ¿Proporcionará el sitio contenido personalizado sin que el usuario deba iniciar la sesión?
- ¿Cuentas de usuario? ¿Se almacenan las cuentas de usuario en las cuentas de dominio de Windows NT, Active Directory, o bien, lo hacen en un almacén de datos como, por ejemplo, una base de datos relacional, un servicio de directorio LDAP (Protocolo ligero de acceso de directorios) alternativo o un archivo XML?
- ¿Es necesario un único inicio de sesión o que éste se realice sin problemas? ¿Se desea que los usuarios inicien la sesión desde una página de inicio de sesión, o bien, es necesario que la autenticación se produzca de forma automática? Por ejemplo, se puede requerir una autenticación automática para una transacción automatizada entre empresas (B2B).
- ¿Se necesita un inicio de sesión seguro? ¿Es necesario reforzar al máximo la seguridad del sistema para evitar los robos en la red de nombres de usuario y contraseñas por parte de los intrusos? Esta decisión se suele tomar según el carácter de confidencialidad de los datos disponibles en el sitio.
- ¿Se ejecutará la aplicación en Internet? ¿Dispondrá la aplicación de un servidor de seguridad, donde los usuarios no se autentican en un dominio, o bien, se basará en una Intranet donde los usuarios finales ya se pueden autenticar en un dominio?
- ¿Se debe delegar el contexto de seguridad? ¿Es necesario que los componentes empresariales se ejecuten con la identidad del autor de la llamada? Si es así, la representación es necesaria. Asimismo, si se necesita tener acceso a los recursos del sistema como, por ejemplo, colas de mensajes, bases de datos o sistema de archivos en equipos remotos, será necesaria la representación a nivel de delegación.
- ¿Ejecutan los servidores y los clientes únicamente Windows 2000? ¿Se dispone de un entorno homogéneo de equipos que ejecutan Windows 2000, o bien, existen clientes que ejecutan otros sistemas operativos como Windows 9x y Windows NT 4.0?

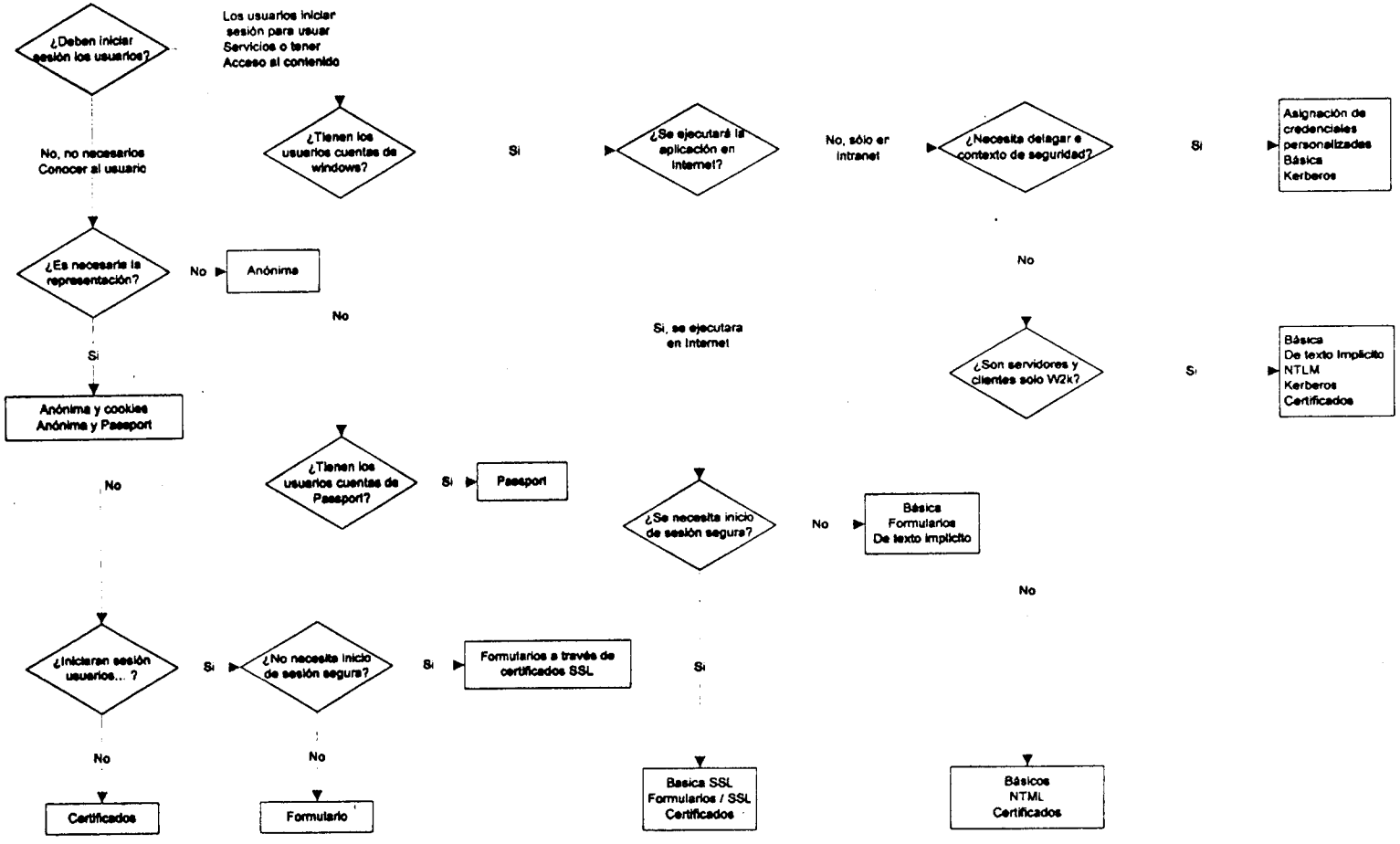



**Figura 3-5 Diagrama de flujo empleado en la selección del método de autenticación más adecuado**

### 3.6.8.2 SEGURIDAD ENTRE LA APLICACIÓN Y LA BASE DE DATOS

En el caso de nuestra aplicación, la cual cuenta con dos niveles físicos o lógicos (En función de los recursos disponibles por nuestro cliente), los usuarios registrados inician una sesión de forma segura en la aplicación basada en Web a través del explorador Web. La aplicación Web establece conexiones seguras con una base de datos a fin de administrar básicamente las tareas de recuperación de datos. Estos escenarios son comunes en las aplicaciones de portal, que ofrecen contenidos de noticias a los suscriptores registrados. La Figura 3-6 es un claro ejemplo.

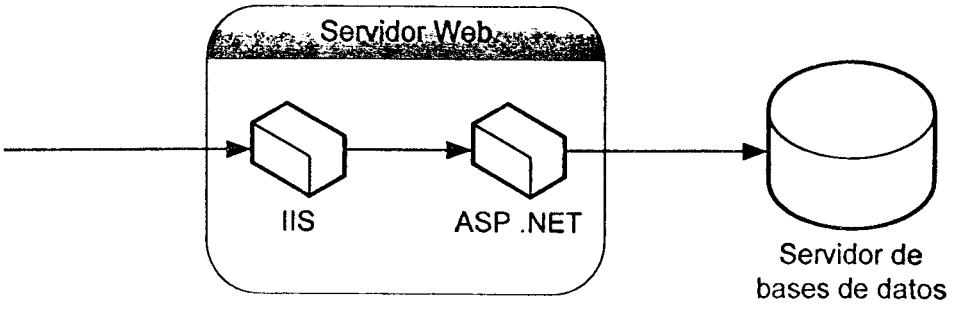



**Figura 3-6 Escenario de Internet de una aplicación Web basada en ASP.NET con conexión a SQL Server**

Este escenario tiene las siguientes características:

- Los usuarios cuentan con distintos tipos de exploradores.
- Los usuarios anónimos pueden explorar las páginas no restringidas de la aplicación.
- Los usuarios deben registrarse o iniciar una sesión (a través de un formulario HTML) para ver las páginas restringidas.
- Las credenciales de usuario se validan con una base de datos.
- Toda la información suministrada por el usuario (como las credenciales) que se utiliza en las consultas a la base de datos se valida a fin de reducir la amenaza de ataques de inyección.
- Se recomienda que la aplicación Web se ubica en una red perimetral también conocida como "DMZ", zona desmilitarizada), con servidores de seguridad que la separan de Internet y la red empresarial interna (y la base de datos).
- La aplicación requiere una seguridad extrema, altos niveles de escalabilidad y un proceso de auditoría exhaustivo.
- La base de datos confía en la aplicación para llevar a cabo correctamente la autenticación de los usuarios (es decir, la aplicación efectúa llamadas a la base de datos en nombre de los usuarios).
- La aplicación Web se conecta a la base de datos mediante la cuenta de proceso ASP .NET.
- Se utiliza una única función de base de datos para la autorización de la base de datos.

Con esta propuesta, la aplicación Web presenta una página de inicio de sesión para aceptar las credenciales. Los usuarios que obtengan la validación tendrán permiso para continuar; el resto verá como se les deniega el acceso. La base de datos lleva a cabo la autenticación mediante la identidad de proceso predeterminada de ASP .NET, una cuenta con privilegios mínimos (es decir, la base de datos confía en la aplicación ASP .NET).

**Tabla 3-7 Resumen de seguridad**

| Categoría | Detalles |
|---|---|
| Autenticación | El servidor Web se configura para permitir el acceso anónimo; la aplicación autentica a los usuarios a través del proceso de autenticación mediante formularios para obtener las credenciales.<br>La validación se efectúa con una base de datos.<br>Las contraseñas de los usuarios no se almacenan en la base de datos. En cambio, sí se almacenan los algoritmos hash de las contraseñas con los valores salt. El valor salt reduce la amenaza asociada a los ataques de diccionario.<br>La autenticación se emplea para establecer la conexión con la base de datos que utiliza la cuenta con privilegios mínimos para ejecutar la aplicación Web. |
| Autorización | La cuenta de proceso utilizada recibe autorización para obtener acceso a los recursos de sistema del servidor Web. Los recursos están protegidos mediante listas de control de acceso<br>La autorización para el acceso a la base de datos se efectúa mediante la identidad de la aplicación. |
| Comunicación segura | Proteja el envío de datos importantes de los usuarios a la aplicación Web mediante SSL.<br>Proteja el envío de datos importantes del servidor Web al servidor de bases de datos mediante IPSec. |

### 3.6.9 ESQUEMA DE SEGURIDAD PROPUESTO

La aplicación de un esquema como el antes mencionado da por resultado la configuración de seguridad recomendada para este escenario (ver la Figura 3-7).

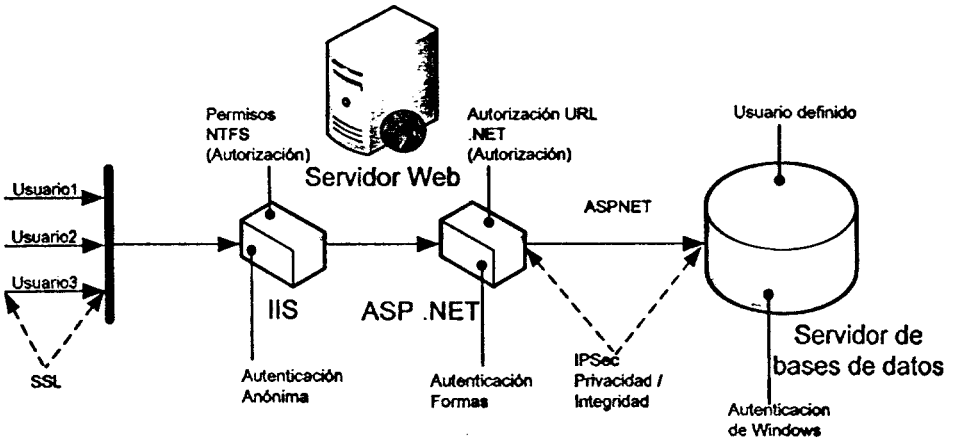



**Figura 3-7 Configuración de seguridad recomendada para el escenario de Internet de ASP.NET a SQL Server**

### 3.6.10 ANÁLISIS

- La autenticación mediante Formularios resulta idónea en el ambiente basado en Web, debido a que los usuarios no disponen de cuentas de usuarios pertenecientes al la estructura de usuarios de nuestro cliente.
- La página de inicio de sesión por formularios se utiliza para obtener las credenciales de usuario, el proceso de validación de dichas credenciales debe realizarse mediante el código de la aplicación, se puede emplear cualquier tipo de datos almacenado. La solución más habitual son las bases de datos.
- La autenticación mediante Formularios exige proteger las credenciales de inicio de sesión iniciales mediante SSL. También es necesario proteger el vale de la autenticación mediante Formularios (El cual se transmite como si se tratara de una "cookie" en las solicitudes Web posteriores del cliente autenticado). Podría utilizarse SSL para todas las páginas a fin de proteger el vale, o bien cifrar el vale de la autenticación.
- Las aplicaciones Web desarrolladas con la tecnología .NET se ejecuta como la cuenta ASPNET local con privilegios mínimos, de modo que se reducen las probabilidades de daños ocasionados por una posible exposición.
- La autorización de direcciones URL en el servidor Web permite que los usuarios no autenticados puedan explorar páginas Web no restringidas además de exigir la autenticación para las páginas restringidas.
- Dado que no está habilitada la suplantación, cualquier acceso a recursos locales o remotos que efectúe la aplicación basada en Web se lleva a cabo mediante el contexto de seguridad de la cuenta ASPNET. Las listas de control de acceso para los recursos seguros deberían configurarse como corresponda.
- Las credenciales de usuario se validan con relación a una base de datos personalizada. Las contraseñas hash (con valor salt) se almacenan en la base de datos.
- El uso de una cuenta duplicada en el servidor de bases de datos (una cuenta que coincida con la cuenta de proceso ASP.NET) se traduce en una mayor carga de administración. Si se cambia una contraseña de un equipo, por ejemplo, es preciso sincronizarla y actualizarla en el resto de los equipos. Algunos escenarios ofrecen la posibilidad de utilizar una cuenta de dominio con privilegios mínimos a fin de agilizar la administración.
- La posibilidad del uso de IPSec entre el servidor Web y el servidor de bases de datos garantiza la privacidad de los datos que se envían desde la base de datos y hacia la misma.
- Se recomienda la implementación de SSL entre el explorador y el Web protege las credenciales y cualquier otro tipo de datos cuya seguridad sea importante.
- Si utiliza una granja de servidores Web, asegúrese de que las claves de cifrado sean coherentes en todos los servidores de la granja.

### 3.6.11 DESVENTAJAS

La aplicación debe transmitir la identidad original del llamador hasta la base de datos para satisfacer los requisitos de auditoría. La identidad del llamador puede transferirse mediante los parámetros de procedimiento almacenado.

## 3.7 RECOMENDACIÓN DE SEGURIDAD ADICIONAL

Nuestra aplicación transmite datos confidenciales en la red desde los usuarios y a los usuarios y entre nodos de aplicaciones intermedios. Entre los datos confidenciales, pueden figurar credenciales utilizadas para la autenticación o información almacenada en la base de datos que nuestro cliente considere confidencial. Para evitar la revelación indeseada de información y proteger los datos contra modificaciones no autorizadas durante la transmisión, deberá protegerse el canal entre los extremos de comunicación.

Esta sección se presenta las dos tecnologías básicas que pueden utilizarse para mantener la confidencialidad y la integridad de mensajes para los datos que se transmiten en la red entre clientes y servidores por Internet e intranets corporativas. Se trata de SSL ("Secure Sockets Layer") e IPSec ("Internet Protocol Secure").

- **SSL (Secure Sockets Layer).** Suele utilizarse para proteger el canal entre un explorador y el servidor Web. No obstante, también puede utilizarse para proteger mensajes de servicios Web y comunicaciones con un servidor de bases de datos.
- **Seguridad del protocolo Internet (IPSec).** IPSec ofrece una solución para la comunicación segura en el nivel de transporte y puede utilizarse para proteger los datos enviados entre dos equipos, como por ejemplo, entre un servidor de aplicaciones y un servidor de bases de datos.

### 3.7.1 SSL

SSL sirve para establecer un canal de comunicación cifrada entre el cliente y el servidor.

#### 3.7.1.1 UTILIZAR SSL

Al utilizar SSL, deberá tener en cuenta lo siguiente:

- Cuando se aplica SSL, el cliente utiliza el protocolo HTTPS (y especifica una dirección URL https://) y el servidor escucha en el puerto TCP 443.
- Deberá supervisar el rendimiento de la aplicación al habilitar SSL. SSL utiliza funciones criptográficas complejas para cifrar y descifrar datos y, por lo tanto, afecta al rendimiento de la aplicación. La mayor disminución del rendimiento se produce durante el protocolo de enlace inicial, en el que se utiliza cifrado de claves públicas y privadas. Posteriormente (una vez generada e intercambiada una clave de sesión segura), se utiliza cifrado simétrico más rápido para cifrar los datos de la aplicación.

- Deberá optimizar las páginas que utilizan SSL; para ello, incluya menos texto y use gráficos más sencillos en las páginas.
- Puesto que el aumento del rendimiento asociado a SSL es mayor durante el establecimiento de la sesión, deberá asegurarse de que no se agote el tiempo de espera de las conexiones.
- SSL requiere que se haya instalado un certificado de autenticación de servidor en el servidor Web (o en el servidor de bases de datos).

### 3.7.2 IPSEC

IPSec puede utilizarse para proteger los datos enviados entre dos equipos, como por ejemplo, un servidor de aplicaciones y un servidor de bases de datos. IPSec es totalmente transparente para las aplicaciones al implementarse los servicios de cifrado, integridad y autenticación en el nivel de transporte. Las aplicaciones siguen comunicándose entre sí de la forma habitual mediante puertos TCP y UDP.

IPSec le permite:

- Proporcionar confidencialidad de mensajes al cifrar todos los datos enviados entre dos equipos.
- Proporcionar integridad de mensajes entre dos equipos (sin cifrado de datos).
- Proporcionar autenticación mutua entre dos equipos (no usuarios). Por ejemplo, puede ayudar a proteger un servidor de bases de datos si establece una directiva que admite peticiones solamente de un equipo cliente específico (por ejemplo, un servidor Web o de aplicaciones).
- Restringir los equipos que pueden comunicarse entre sí. También puede limitar la comunicación a protocolos IP y puertos TCP/UDP específicos.

#### 3.7.2.1 UTILIZAR IPSEC

Al utilizar IPSec, deberá tener en cuenta lo siguiente:

- IPSec puede utilizarse tanto para la autenticación como para el cifrado.
- Los desarrolladores no disponen de API de IPSec para controlar la configuración mediante programación. Toda la supervisión y configuración de IPSec se realiza en el complemento IPSec, mediante la directiva de seguridad local de Microsoft.
- IPSec no puede proteger todos los tipos de tráfico IP en el sistema operativo Microsoft Windows® 2000.En concreto, no se puede utilizar para proteger el tráfico de difusión, multidifusión, intercambio de claves de Internet o Kerberos (que ya es de por sí un protocolo seguro).
- Los filtros IPSec sirven para controlar cuándo se aplica IPSec. Para probar las directivas IPSec, utilice el Monitor de IPSec. El Monitor de IPSec (Ipsecmon.exe) proporciona información acerca de qué directiva IPSec está activa y de si se ha establecido un canal seguro entre equipos.
- Para establecer una confianza entre dos servidores, puede utilizar IPSec con autenticación mutua. Ésta utiliza certificados para autenticar ambos equipos.

- Si necesita utilizar IPSec para proteger la comunicación entre dos equipos que están separados por un servidor de seguridad, asegúrese de que el servidor de seguridad no utiliza la Traducción de direcciones de red (NAT, Network Address Translation). IPSec no funciona con ningún dispositivo basado en NAT.

## 3.8 IDENTIFICACIÓN DEL LOS RECURSOS A PROTEGER

Cuando una petición Web se transmite por los niveles de implementación físicos de la aplicación, cruza varios canales de comunicación. La Figura 3-8 muestra un modelo de implementación de aplicaciones Web utilizado a menudo.

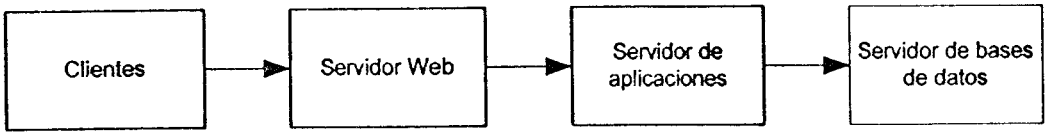



**Figura 3-8 Un modelo típico de implementación Web**

En este modelo de implementación, una solicitud pasa por tres canales distintos. El vínculo del cliente al servidor Web puede estar en Internet o en la intranet corporativa y suele utilizar HTTP. Los dos vínculos restantes se realizan entre servidores internos del dominio corporativo. No obstante, los tres vínculos acarrean posibles problemas de seguridad. Muchas aplicaciones basadas exclusivamente en intranet transmiten datos confidenciales de un nivel a otro.

La Figura 3-9 muestra cómo se puede proteger cada canal mediante una combinación de SSL, IPSec.

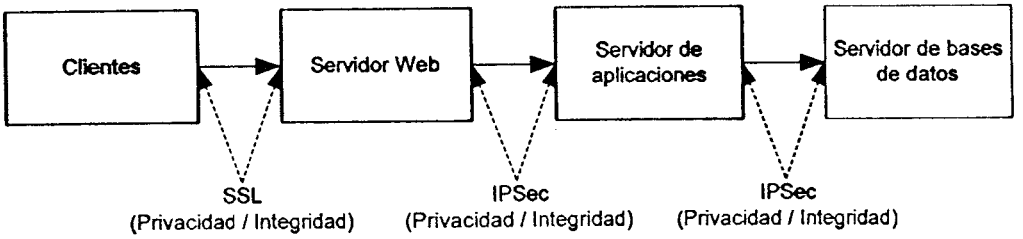



**Figura 3-9 Un modelo típico de implementación Web, con comunicaciones seguras**

La tecnología elegida depende de varios factores tales como el protocolo de transporte, las tecnologías empleadas por los equipos en los extremos del canal de comunicación y las consideraciones del entorno (como el hardware, las versiones del sistema operativo, los servidores de seguridad, etc.).

### 3.8.1 SEGURIDAD PUNTO A PUNTO

Los escenarios de comunicación punto a punto pueden clasificarse en los siguientes temas:

- Explorador y servidor Web
- Servidor de aplicaciones y servidor de bases de datos

### 3.8.2 EXPLORADOR Y SERVIDOR WEB

Para proteger datos confidenciales enviados entre un explorador y un servidor Web, deberá utilizar SSL. Necesitará utilizar SSL en las siguientes situaciones:

- Utiliza la autenticación mediante Formularios y necesita proteger las credenciales de texto no cifrado enviadas a un servidor Web desde un formulario de inicio de sesión. En este escenario, deberá utilizar SSL para proteger el acceso a todas las páginas (y no sólo la página de inicio de sesión) con el fin de garantizar que la cookie de autenticación que se genera a partir del proceso de autenticación inicial permanezca segura durante toda la duración de la sesión de explorador del cliente con la aplicación.
- Utiliza la autenticación básica y necesita proteger las credenciales de texto no cifrado. Deberá utilizar SSL para proteger el acceso a todas las páginas (y no sólo la página de inicio de sesión), puesto que la autenticación básica envía las credenciales de texto no cifrado al servidor Web con todas las peticiones de la aplicación (y no sólo la inicial).
- Su aplicación transmite datos confidenciales del explorador al servidor Web (y viceversa), que en nuestro caso puede ser la identificación del usuario.

#### 3.8.2.1 SERVIDOR DE APLICACIONES Y SERVIDOR DE BASES DE DATOS

Para proteger los datos enviados entre un servidor de aplicaciones y un servidor de bases de datos, puede utilizar IPSec. Si el servidor de bases de datos ejecuta SQL Server 2000 (y las bibliotecas de red de SQL Server 2000 están instaladas en el servidor de aplicaciones), puede utilizar SSL. Esta última opción requiere que esté instalado un certificado de autenticación de servidor en el almacén del equipo del servidor de bases de datos.

Es posible que necesite proteger el vínculo al servidor de bases de datos en las siguientes situaciones:

- Se conecta al servidor de bases de datos y no utiliza la autenticación de Windows. Por ejemplo, utiliza la autenticación de SQL para SQL Server o se conecta a una base de datos que no es de SQL Server. En estos casos, las credenciales se transmiten como texto no cifrado, lo que puede acarrear un problema de seguridad considerable.
- Su aplicación envía datos confidenciales a la base de datos y también recupera datos confidenciales de la base de datos.

### 3.8.2.2 UTILIZAR SSL PARA SQL SERVER

Considere los siguientes aspectos si utiliza SSL para proteger el canal a una base de datos de SQL Server:

- Para que funcione SSL, debe instalar un certificado de autenticación de servidor en el almacén del equipo servidor de bases de datos. El equipo cliente deberá tener además un certificado de entidad emisora raíz de la misma entidad que emitió el certificado de servidor (o una entidad de confianza).
- Los clientes deben tener instaladas las bibliotecas de conectividad de SQL Server 2000. Las versiones anteriores de bibliotecas genéricas no son válidas en este caso.
- SSL sólo funciona con TCP/IP (el protocolo de comunicación recomendado para SQL Server) y las canalizaciones con nombre.
- Puede configurar el servidor de forma que exija el uso de cifrado para todas las conexiones (de todos los clientes).
- En el cliente, puede:
- Exigir el uso de cifrado para todas las conexiones salientes.
- Permitir que las aplicaciones cliente puedan elegir si utilizan o no el cifrado en cada conexión mediante la cadena de conexión.
- A diferencia de IPSec, en SSL los cambios de configuración no son necesarios si se modifican las direcciones IP del cliente o del servidor.

### 3.8.3 ELEGIR ENTRE IPSEC Y SSL

Considere los siguientes aspectos a la hora de elegir entre IPSec y SSL:

- IPSec puede utilizarse para proteger todo el tráfico IP entre equipos; SSL es específico de una aplicación concreta.
- IPSec es una configuración general del equipo y no admite el cifrado de conexiones de red específicas. No obstante, los sitios pueden dividirse para utilizar o no utilizar SSL. Además, cuando se utiliza SSL para establecer la conexión con SQL Server, puede elegir si desea utilizar o no SSL en cada conexión (desde la aplicación cliente). SSL puede funcionar a través de un servidor de seguridad basado en NAT e IPSec no. IPSec requiere que ambos equipos ejecuten Windows 2000 o posterior.
- IPSec es transparente para las aplicaciones y, por lo tanto, puede utilizarse con protocolos seguros que se ejecutan sobre IP, como HTTP, FTP y SMTP. Por el contrario, SSL está vinculado estrechamente a la aplicación.
- IPSec, además de para el cifrado, también puede utilizarse para la autenticación de equipos. Resulta especialmente importante para los escenarios de subsistemas de confianza, en los que la base de datos autoriza a una identidad fija desde una aplicación específica (que se ejecuta en un equipo determinado). IPSec puede utilizarse para garantizar que sólo el servidor de aplicaciones específico pueda conectarse al servidor de bases de datos, con el fin de evitar ataques desde otros equipos.

### 3.8.4 AUDITORIA INFORMÁTICA

#### 3.8.4.1 JUSTIFICACIÓN PARA EFECTUAR UNA AUDITORÍA DE SISTEMAS

Aun cuando el ahorro de recursos humanos y económicos es suficiente razón para iniciar un proceso de auditoría, existen aspectos que hacen de la auditoría un proceso de revisión necesario para mantener al día la información y rendimiento real de un área día con día más indispensable en las empresas. De forma general se presentan algunos beneficios adicionales, para tomar en cuenta la realización de una auditoría.

- Evitar el aumento considerable e injustificado del presupuesto asignado al departamento de informática.
- Evitar el desconocimiento en el nivel directivo de la situación informática de la empresa.
- Corregir la Falta total o parcial de seguridad lógica y física, promoviendo que se garantice la integridad del personal, equipos e información.
- Prevención o descubrimiento de fraudes efectuados con el sistema.
- Corregir la falta de una planificación informática.
- Solucionar o prevenir que existan áreas que no funcionan correctamente, carentes de políticas, objetivos, normas, metodología, asignación de tareas y adecuada administración del recurso humano.
- Descontento general de los usuarios por incumplimiento de plazos y mala calidad de los resultados, reflejo de la mala organización de las diversas áreas de informática de la empresa.
- Documentación de sistemas nula o incompleta que revela la dificultad de efectuar el mantenimiento de los sistemas en producción.

## 3.9 MARCO JURÍDICO PARA LA PROTECCIÓN DE LOS PRODUCTOS INFORMÁTICOS

### 3.9.1 LA PROTECCIÓN JURÍDICA DE LOS PROGRAMAS DE COMPUTACIÓN

Antes de hablar de su protección jurídica veamos como la Ley federal del derecho de autor, define a los programas de computadoras.

Según el Título IV de dicha ley en su Capítulo IV; "De los Programas de computación y las Bases de Datos", define; en el Artículo 101. "Se entiende por programa de computación la expresión original en cualquier forma, lenguaje o código, de un conjunto de instrucciones que, con una secuencia, estructura y organización determinada, tiene como propósito que una computadora o dispositivo realice una tarea o función específica".

Un programa de computadora es un bien inmaterial, ya que tiene las siguientes características: es fruto o creación de la mente, para que se haga perceptible para el mundo exterior, es necesario plasmarlo en un soporte, motivos por los cuales se debe procurar su protección jurídica. La protección jurídica de los programas de computadoras, se puede

implementar utilizando las siguientes instituciones jurídicas conocidas: estipulaciones contractuales, secreto comercial, derecho de patentes, derecho de marcas y derecho de autor. La protección de los programas de computadoras está regulada por la Ley federal del derecho de autor.

### 3.9.2 LOS DELITOS INFORMÁTICOS

Se define el delito informático como toda acción culpable realzada por un ser humano, que cause un perjuicio a personas sin, que necesariamente se beneficie el autor o que, por el contrario, produzca un beneficio ilícito a su autor aunque no perjudique de forma directa o indirecta a la victima, tipificado por la ley.

El delito informático se puede dividir en grupos de figuras delictivas claramente diferenciadas:

- Delitos contra la intimidad.
- Delitos contra el patrimonio.
- Falsedades documentales.

Los delitos contra el patrimonio que estén relacionados con los derechos legales de comercialización y uso, se encuentran previstos por el Capítulo IV artículos 102 al 114 de la ley federal del derecho de autor.

Para ejemplifican la gravedad e impacto en la sociedad, se listan solo algunos de los actos más representativos de los delitos contra la intimidad, el patrimonio de terceros y las falsedades documentales,

- **Delitos contra la intimidad:** (Apoderarse de mensajes electrónicos, documentos electrónicos, interceptación de comunicaciones, utilización de artificios técnicos para escuchar, transmitir, grabar o reproducir sonido, imagen o cualquier otra señal de comunicación).
- **Delitos contra el patrimonio:** (Llaves de tarjetas magnéticas o instrumentos de apertura a distancia).
- **Estafas informáticas:** (Transferencias no consentidas de cualquier activo patrimonial en perjuicio de un tercero).
- **Defraudaciones:** (Uso de cualquier equipo terminal de comunicación).
- **Daños informáticos:** (Aquel que por cualquier medio destruya, altere, inutilice o dañe datos, programas o documentos electrónicos ajenos en redes, soportes o sistemas, incluye virus, bombas lógicas y hackers).
- **Propiedad intelectual:** (Quien con ánimo de lucro reproduzca, plagie, distribuya o comunique públicamente, en todo o en parte, una obra literaria, artística o científica en cualquier tipo de soporte o comunicada a través de cualquier medio sin autorización, en si, la piratería informática).
- **Delitos de falsedades:** (Todo documento que soporte material que exprese o incorpore datos con eficacia probatoria o cualquier tipo de relevancia jurídica, aún

cuando el documento sea electrónico, como falsificación de tarjetas de débito y crédito).

En materia de legislativa, México no cuenta con leyes bastas y suficientes que ofrezcan un marco legal para proteger a las personas sobre este tipo de delitos. El estado de Sinaloa incluye en su "Código Penal para el Estado de Sinaloa", el Titulo Décimo, Capitulo V, Artículo 217, representa el primer esfuerzo por contar con una ley que garantice el contenido y uso de los datos almacenados por un sistema informático.

### 3.9.3 DEFINICIÓN DE AUDITORÍA

- **DEFINICIÓN GENÉRICA:** Es el examen constructivo y preventivo sobre eventos que ocurren en el entorno social, con el objeto de promover la eficiencia de su ejecución y el uso óptimo de los recursos disponibles.
- **DEFINICIÓN DEL "I.M.C.P.":** Es una actividad profesional que requiere de una técnica especializada o destreza y la aceptación de una responsabilidad pública.

Patín etal., (1998), presentan una definición integral; Es la actividad consistente en revisar sistemáticamente, si el objeto sometido a análisis representa adecuadamente la realidad que pretende reflejar y/o cumple las condiciones que le han sido prescritas. Culminando con la emisión de una opinión profesional llamada dictamen.

### 3.9.4 TIPOS DE AUDITORÍA.

Con la especialización de los objetos a auditar y la responsabilidad social ligada a la emisión de un dictamen ha sido necesario especializar al personal involucrado en la auditoría, así mismo fue necesario crear especialidades dentro de la auditoria. El "I.M.C.P." presenta un listado con las principales especializaciones de la auditoría.

- AUDITRÍAS ESPECIALES.
- AUDITORÍA ADMINISTRATIVA
- AUDITORÍA OPERACIONAL
- AUDITORÍA EN INFORMÁTICA
- AUDITORÍAS DETALLADAS.
- AUDITORÍA DE ESTADOS FINANCIEROS.

Las necesidades del proyecto requieren de la planeación de una auditoria en informática, la cual es desarrollada a lo largo del presente capitulo.

### 3.9.5 DEFINICIÓN DE AUDITORÍA DE SISTEMAS DE INFORMACIÓN

Es el proceso de recolección y evaluación de evidencia para determinar si un sistema automatizado cumple con las expectativas depositadas por el cliente, Las actividades comúnmente asignadas al área de informática se presentan en la Tabla 3-8.

Con una auditoría de sistemas se puede garantizar la calidad de la información existente en las bases de datos de los sistemas informáticos que se utilizan para controlar los recursos, su entorno y los riesgos asociados a esta actividad. La información recabada en una auditoría da la posibilidad de iniciar el análisis de los aspectos relativos a las bases de datos de los sistemas informáticos en que se haya detectado algún tipo de alteración o incorrecta operación de las mismas. Una auditoría puede extender sus alcances hasta llegar a la revisión de los planes de medidas elaborados en auditorías informáticas anteriores, debido a un dictamen con opinión denegada o abstención de opinión. El tema referente al dictamen será revisado en la sección 3.9.22.2.

**Tabla 3-8 Actividades comúnmente asignadas al área de informática**

| Objetivos de la auditoría | Eventos a prevenir |
|---|---|
| Salvaguarda de activos | Daños<br>Destrucción<br>Uso no autorizado<br>Robo |
| Integridad de los datos | Oportuna<br>Precisa<br>Confiable<br>Completa |
| Garantizar el cumplimiento de las metas de la organización | Contribución al desempeño y productivaza de las funciones naturales de la organización |
| Uso eficientemente de los recursos | Utiliza los recursos adecuadamente en el procesamiento de la información |

*Fuente: López, 2001*

### 3.9.6 OBJETIVOS GENERALES DE UNA AUDITORÍA DE SISTEMAS

El principal objetivo de una auditoria de sistemas, es la reducción de los gastos indirectos, producto de fallas y contratiempos originados por; incidentes de seguridad tanto internos como externos, escasez de recursos humanos y económicos, deficiencias en manuales de procedimiento y planes de recuperación de desastres.

La auditoría de sistemas, busca identificar las áreas que requieran inversión de recursos tanto humanos como económicos, que deriven en incremento de la productividad del resto del personal. Otros objetivos que busca la auditoría se listan a continuación:

- Buscar una mejor relación costo-beneficio de los sistemas automáticos o computarizados diseñados e implantados.
- Incrementar la satisfacción de los usuarios del sistema, por medio de la revisión de objetivos y metodologías de los departamentos que componen el área de sistemas o informática

- Asegurar una mayor integridad, confidencialidad y confiabilidad de la información mediante la recomendación de seguridades y controles.
- Conocer la situación actual del área informática y las actividades y esfuerzos necesarios para lograr los objetivos propuestos por la misma.
- Seguridad de personal, datos, hardware, software e instalaciones, actualización de los manuales de procedimientos y planes de contingencia.
- Minimizar existencias de riesgos en el uso de Tecnología de información.
- Decisiones de inversión y gastos innecesarios.
- Capacitación y educación sobre controles en los Sistemas de Información.

## 3.9.7 CLASES DE AUDITORÍA DE SISTEMAS DE INFORMACIÓN

### 3.9.7.1 AUDITORÍA INTERNA Y AUDITORÍA EXTERNA

La Auditoría interna es realizada con recursos materiales y humanos que pertenecen a la empresa auditada. Los empleados que realizan esta tarea son remunerados económicamente. El departamento de auditoría interna existe por expresa decisión de la empresa, o sea, que puede optar por su disolución en cualquier momento.

La Auditoría externa es realizada por personal externo a la empresa auditada; es siempre remunerada. Se presupone una mayor objetividad que en la Auditoría Interna, debido al mayor distanciamiento entre auditores y auditados. Una institución que posee un departamento de Auditoría Interna puede y debe en ocasiones contratar servicios de Auditoría Externa. Las razones para hacerlo suelen ser:

- Necesidad de auditar una materia de gran especialización, para la cual los servicios propios no están suficientemente capacitados.
- Contrastar algún Informe interno con el que resulte del externo, en aquellos supuestos de emisión interna de graves recomendaciones que chocan con la opinión generalizada de la propia empresa.
- Servir como mecanismo protector de posibles Auditorías informáticas externas decretadas por la misma empresa.
- Aunque la Auditoría interna sea independiente del Departamento de Sistemas, sigue siendo la misma empresa, por lo tanto, es necesario que se le realicen Auditorías externas como para tener una visión desde afuera de la empresa.

Para los fines de nuestra propuesta, se recomienda optar por una auditoria interna, evitando así impactar de forma importante en los recursos económicos de nuestro cliente y permitirá ajustar los esquemas y procedimientos de nuestro departamento de auditoría interna. Los resultados de los primeros ejercicios de auditoría informática, se recomienda que sean cotejados con los resultados de un ejercicio de auditoria externa, de esta forma se podrá determinar el grado de confianza de nuestra auditoría interna.

### 3.9.8 FUNCIÓN DE LA AUDITORÍA DE SISTEMAS DE INFORMACIÓN

En este punto es necesario recordar que la veracidad y calidad de una auditoría esta relacionada con la disposición de los departamentos involucrados en la misma. La sección 3.9.22.2, se habla del informe con abstención de opinión, el cual es producto de la poca o nula información recabada de uno o varios rubros revisados por la auditoría. Con ello se impide la correcta evaluación de los objetivos fijados al inicio de la auditoría y por consiguiente se obliga a dictaminar con abstención de opinión. La función de Auditoría Informática debe realizar varias actividades objetivas, algunas de ellas son:

- Verificación del control interno, tanto de las aplicaciones como de los sistemas informáticos, centrales y periféricos.
- Análisis de gestión de los sistemas informáticos desde un punto de vista de riesgo, de seguridad, de gestión y de efectividad de la gestión.
- Análisis de la integridad, fiabilidad y certeza de la información a través del análisis de las aplicaciones.
- Auditoría del riesgo operativo de los canales de información.
- Análisis de la gestión de los riesgos de la información y de la seguridad implícita
- Verificación del nivel del arte tecnológico de la instalación revisada y de las consecuencias empresariales que un retraso tecnológico puede acarrear.
- Diagnóstico sobre el grado de cobertura que dan las aplicaciones a las necesidades estratégicas y operativas de información de la organización.

La importancia clara de la auditoría es que permite prevenir y corregir oportunamente eventos desfavorables para quien se somete a una auditoría. La verificación al estado actual que presenta la infraestructura informática, así como la revisión a los esquemas de crecimiento, permiten vigilar el rumbo del área de informática, otras actividades van enfocadas a verificar la aplicación correcta y precisa de procedimientos y recomendaciones, producto de auditorías pasadas.

### 3.9.9 ORGANIZACIÓN DE LA FUNCIÓN DE AUDITORÍA INFORMÁTICA

La organización de la auditoría de sistemas de información, es un proceso muy delicado que debe ser considerado adecuadamente, la naturaleza de un departamento de esta clase requiere una adecuada localización dentro del organigrama, por lo general siempre se encuentra en niveles staff, así se garantizar la presencia de controles necesarios y la suficiente independencia para cumplir con su labor. La organización de la auditoría de sistemas de información debe contemplar los siguientes principios:

- Su localización dentro del organigrama puede estar ligada a la posición que ocupa la auditoría interna operativa y financiera, pero con independencia de objetivos (aunque haya una coordinación lógica entre ambos departamentos), de planes de formación y de presupuestos.
- La organización operativa debe de ser la de un grupo independiente del de Auditoría interna, con una accesibilidad total a los sistemas informáticos e idealmente

dependiendo de la misma persona en la empresa que la Auditoría interna, que debería ser el director general.

- La dependencia, en todo caso, debe ser del máximo responsable operativo de la organización, nunca del departamento de sistemas, ni del departamento financiero y/o administrativo.
- La gestión de la función debe ser llevada a cabo por personal que haya o esté trabajando en Auditoría informática.
- Los recursos humanos con los que debe contemplar una mezcla equilibrada entre personas con formación en Auditoría y organización y personas con perfil informático.

Una propuesta de implementación de la organización interna de las funciones de la auditoría de sistemas de información podría ser:

- **Jefe de departamento:** Desarrolla el plan operativo del departamento, las descripciones, los puestos de trabajo del personal a su cargo, las planificaciones de actuación a un año, los cambios en los métodos de trabajo y evalúa la capacidad de personas a su cargo.
- **Gerente o Supervisor de Auditoría informática:** Trabaja estrechamente con el jefe del departamento en las tareas operativas diarias, ayuda en la evaluación del riesgo de cada uno de los trabajos, realiza los programas de trabajo, dirige y supervisa directamente a las personas en cada uno de los trabajos de los que es responsable.
- **Auditor Informático:** Son responsables de la ejecución directa del trabajo, deben tener una especialización genérica pero también una específica. Su trabajo consistirá en la obtención de información, ejecución de pruebas, documentación del trabajo, evaluación y diagnóstico de resultados.

### 3.9.10 METODOLOGÍAS DE EVALUACIÓN DE SISTEMAS

En función de los objetivos que se desean alcanzar por medio de una auditoría informática, se empleara una de las dos metodologías existentes. La metodología de "ANÁLISIS DE RIESGOS" cuya finalidad es facilitar la "evaluación" de los riesgos y ofrece recomendaciones de acciones basadas en el costo-beneficio de las mismas; Por el contrario la metodología de "AUDITORÍA INFORMÁTICA" tiene por finalidad identificar el nivel de "exposición" por la falta de controles. Para ambas metodologías, se exponen algunas definiciones empleadas por ambas metodologías:

- **AMENAZA:** Una(s) persona(s) vista(s) como posible fuente de peligro o catástrofe.
- **VULNERABILIDAD:** La situación creada, por la falta de uno o varios controles, con la que la amenaza pudiera acaecer y así afectar al entorno informático.
- **RIESGO:** La probabilidad de que una amenaza llegue a acaecer por una vulnerabilidad.
- **EXPOSICIÓN O IMPACTO:** La evaluación del efecto del riesgo.

### 3.9.10.1 TIPOS DE ANÁLISIS

El uso de una metodología en particular requiere de seleccionar un paradigma de análisis para interpretar la información recolectada. Por tal motivo se puede emplear el análisis cualitativo o cuantitativo. La elección de un paradigma de análisis en particular se encuentra basada en reglas establecidas por el departamento de auditoría, también es posible que la elección se base en la experiencia del jefe de departamento, o este ligada al tipo de empresa a la cual se auditara. La elección del tipo de análisis requiere conocer las características de cada uno de ellos, definiremos al análisis cuantitativo y cualitativo como:

- **Cuantitativo:** Basado en modelos estadísticos y matemáticos, define los procesos por medio de índices, patrones y manuales de procedimientos.
- **Cualitativo:** Basado en el criterio y raciocinio humano, es capaz de definir un proceso de trabajo basándose únicamente en la experiencia acumulada.

#### 3.9.10.1.1 ANÁLISIS CUANTITATIVO

Producirá una lista de riesgos que pueden compararse entre sí con facilidad por tener asignados unos valores numéricos. Estos valores son datos de probabilidad de ocurrencia (riesgo) de un evento que se debe extraer de un registro de incidente donde el número de incidencias tienda al infinito.

#### 3.9.10.1.2 ANÁLISIS CUALITATIVO / SUBJETIVO

Basados en la experiencia y lógica humana. Precisan de la intervención de un profesional experimentado (Ver Tabla 3-9).

**Tabla 3-9 Comparativa entre análisis cuantitativos y cualitativos**

| | Cuantitativa | Cualitativa / Subjetiva |
|---|---|---|
| Pros | Enfoca pensamientos mediante el uso de números. Facilita la comparación de vulnerabilidades muy distintas. Proporciona una cifra "justificante" para cada contramedida. | Enfoque, lo amplio que se desee. Plan de trabajo flexible y reactivo. Se concentra en la identificación de eventos. Incluye factores intangibles. |
| Contras | Estimación de probabilidad depende de estadísticas fiables inexistentes. Estimación de las pérdidas potenciales solo si son valores cuantificables. Metodologías estándares. Difíciles de mantener o modificar. Dependencia de un profesional. | Depende fuertemente de la habilidad y calidad del personal involucrado. Puede excluir riesgos significantes desconocidos. Identificación de eventos reales más claros al no tener que aplicarles probabilidades complejas de calcular. Dependencia de un profesional. |

*Fuente: López, 2001*

### 3.9.11 METODOLOGÍAS DE ANÁLISIS DE RIESGOS

La primera etapa se encuentra basada en la información obtenida con el empleo de cuestionarios, a través de la información obtenida se busca: identifican algunas vulnerabilidades y riesgos, evaluar el impacto, en una etapa posterior el trabajo se enfoca en identificar las contramedidas y el costo. La siguiente etapa es las más importante, con el empleo de una simulación (que comúnmente es llamada "¿QUE PASA SI?...") se analiza el

efecto de las distintas contramedidas en la disminución de los riesgos analizados, eligiendo de esta manera un plan de contramedidas (plan de seguridad) que pondrá el informe final de la evolución.

En la Tabla 3-10 se presentan las actividades necesarias para emplear la metodología de análisis de riesgo.

**Tabla 3-10 Esquema de actividades básicas de la metodología de análisis de riesgo**

| Procedimiento | Etapa 1 | Etapa 2 | Etapa 3 | Etapa 4 | Etapa 5 | Etapa 6 |
|---|---|---|---|---|---|---|
| Cuestionario | X | | | | | |
| Identificar los riesgos | | X | | | | |
| Calcular el impacto | | | X | | | |
| Identificar las contramedidas y el costo | | | | X | | |
| Simulaciones | | | | | X | |
| Elaboración del informe final | | | | | | X |

*Fuente: López, 2001*

## 3.9.12 METODOLOGÍAS DE CLASIFICACIÓN DE LA INFORMACIÓN

El análisis de riesgos no es suficiente, dado que todas las entidades de información a proteger no tienen el mismo grado de importancia y el análisis de riesgos metodológicamente no permite aplicar una diferenciación de contramedidas según el activo o recursos que protege, sino por la probabilidad del riesgo analizado. La clasificación de la información es la herramienta adecuada, el resultado de aplicar una clasificación de la información se plasma en la siguiente oración: "Si identificamos distintos niveles de contramedidas para distintas entidades de información con distinto nivel de criticidad, estaremos optimizando la eficiencia de las contramedidas y reduciendo los costos de las mismas".

### 3.9.12.1 JERARQUÍA DE LA INFORMACIÓN

Información PRIMA, aunque permite definir a voluntad cualquier jerarquía de la información , permitiendo tener un soporte justo a la medida de la empresa que se encuentre auditada, la información PRIMA básicamente define:

- Estratégica (información restringida, confidencial, vital para la subsistencia de la empresa)
- Restringida (a los propietarios de la información)
- De uso interno (a todos los empleados)
- De uso general (sin restricción)

La metodología de la clasificación de la información cuenta con su propia metodología, las etapas descritas en la Tabla 3-11 son las mínimas necesarias para realizar un estudio de la información, cada etapa puede sufrir variaciones para ajustarse al tipo de información que se desea clasificar.

**Tabla 3-11 Metodología clásica de la clasificación de la información**

| Etapa | Descripción |
|---|---|
| Identificación de la información | Basado en la Información PRIMA, se puede categorizar la información recabada por las técnicas de Auditoria ( Sección 0) |
| Inventario de entidades de información residente y operativa. | Inventario de programas, bases de datos, estructuras complejas de datos, soportes de información, entre otros. |
| Identificación de propietarios de la información. | Se realiza una clasificación de los usuarios basado en: Nivel de confidencialidad de la información, y custodian la información. |
| Definición de la matriz de clasificación. | Consiste en definir las políticas, estándares objetivos de control y contramedidas por tipos y jerarquías de información. |
| Confección de la matriz de clasificación. | En esta fase se complementa toda la matriz, asignándole a cada entidad un nivel de jerarquía, lo que la asocia una serie de hitos a cumplir según el punto anterior, para cuyo cumplimiento deberemos desarrollar acciones concretas en el punto siguiente. |

*Fuente: López, 2001*

### 3.9.13 TÉCNICAS DE AUDITORÍA

Debidamente documentada la metodología que se desea emplear para el proceso de auditoría, es necesario describir las técnicas empleadas en la recolección de la información necesaria para cumplir con los objetivos que nos hemos planteados al inicio de la auditoría.

#### 3.9.13.1 TÉCNICAS MANUALES

Estas técnicas son en esencia las mismas que se aplican en otros tipos de Auditoría u otras actividades que requieren una labor de análisis y/o evaluación. Las técnicas de auditoría más utilizadas son presentadas en la

Tabla 3-12, cada una de ellas puede ser utilizada por separado o en conjunto durante una auditoría, los resultados en cuanto a la cantidad y calidad de información recabada por una técnica en específico pueden variar, esto debido a la naturaleza de la empresa auditada.

**Tabla 3-12 Listado de técnicas manuales de recolección de información**

| Técnicas de Inspección manual | Descripción |
|---|---|
| INSPECCIÓN DOCUMENTAL | Esta técnica se utiliza para analizar los procedimientos de control que implican documentación, ya sea la documentación del propio procedimiento o la documentación de sus resultados. Un caso típico de uso es la inspección de manuales, actas de comités, contratos, documentación de sistemas, planes de contingencias, entre otros. Es importante indicar que esta técnica debe utilizarse conjuntamente con la de análisis, es decir, de poco o nada serviría comprobar la existencia de un documento si no se efectúa un análisis de lo completo, actualizado y correcto de su contenido y otras características. |
| ENTREVISTAS | Consiste simplemente en la obtención de información mediante la entrevista a personal que posea conocimiento o experiencia de interés para los objetivos del proyecto de auditoría. |
| ENCUESTAS | Su propósito es la obtención de información mediante la aplicación de cuestionarios predefinidos a un grupo de personas sobre un aspecto en particular. Las personas seleccionadas no necesariamente son parte de la empresa auditada, ya que pueden ser clientes o proveedores, entre otros. Típicamente, esta técnica es utilizada para determinar niveles de satisfacción respecto al servicio proporcionado por un sistema, área de sistemas o por los servicios prestados. |
| SESIONES COORDINADAS | Consiste en el desarrollo de una sesión de trabajo en la cual un coordinador aplica técnicas específicas (técnicas de facilitación) para obtener información de un grupo de expertos en algún tema. El coordinador no tiene opinión sobre el tema tratado, su objetivo es ordenar el trabajo del grupo y facilitar la obtención de información o la generación de productos de trabajo predefinidos. Normalmente, los productos generados implican la clasificación de información, asignación de prioridades o simplemente la generación de nuevas ideas. Este tipo de técnica puede emplearse para validar elementos de la estrategia de la empresa, soluciones a problemas específicos o para obtener conocimientos sobre segmentos de procesos que no se encuentran debidamente documentados o en los cuales participan varios departamentos. |
| CERTIFICACIÓN | Esta es una técnica el juicio del auditor se complementa con la opinión de uno o varios expertos. Este tipo de técnica puede utilizarse en situaciones en que se requiere conocimiento detallado de alguna tecnología en particular o sobre aspectos legales o actuariales. |
| CONFIRMACIÓN | Esta técnica se utiliza para confirmar información que la empresa presenta, mediante la respuesta de una persona ajena a la compañía, como puede ser un cliente, un proveedor o un acreedor. Cuando esta confirmación es masiva, se le conoce como "circulación", es decir, mandar una carta a todos los clientes o a una parte de ellos solicitando que confirmen su saldo. |
| TÉCNICAS DE INGENIERÍA DE INFORMACIÓN | Estas técnicas se emplean tanto en la evaluación de controles generales como en los controles específicos. Evidentemente, su empleo requiere de conocimientos de tecnología de información, mismos que serán necesarios para evaluar los sistemas en sí y los procesos de desarrollo, mantenimiento e integración de paquetes, entre otros. Estas técnicas han sido incluidas bajo la clasificación de "técnicas manuales" porque no es indispensable un soporte automatizado para su aplicación. Sin embargo, es muy probable que se apliquen mediante el uso de herramientas CASE (computer aided software engineering, Ingeniería Asistida por Computadora). |

*Fuente: López, 2001*

### 3.9.13.2 TÉCNICAS APLICADAS A LA TECNOLOGÍA

Estas técnicas son aplicadas en un ambiente de tecnología con el propósito de evaluar el funcionamiento de los componentes tecnológicos del control interno y no sus productos o resultados. Algunos de los ejemplos de este tipo de técnicas serían: la evaluación de la capacidad para manejar altos volúmenes de transacciones de un computador, la velocidad y consistencia de transmisión de un canal de comunicación, la efectividad de una planta de

energía alterna, la efectividad del equipo de detección y extinción de fuego y la eficiencia y exactitud de lectores ópticos.

### 3.9.14 TÉCNICAS ASISTIDAS POR TECNOLOGÍA

Utilizan a la tecnología como una herramienta, pero su objetivo no es evaluar a la tecnología en sí. Su función es evaluar la eficiencia del control interno mediante el examen de sus productos o resultados; esto se aplica tanto a revisiones de controles generales como de controles específicos. Estas técnicas son conocidas por sus siglas en inglés como CAATs (computer assisted audit techniques, Técnicas de Auditoría Asistidas por Computadoras) y pueden ser implementadas básicamente de dos formas.

Utilizando el equipo de cómputo e instalaciones de la propia empresa, con el empleo de utilería y programas de diversas funciones, entre ellos se pueden emplear: generadores de reportes, lenguajes de programación o software especializado de auditoría.

Utilizando paquetes de auditoría que se ejecutan en computadoras personales pertenecientes al grupo de auditoría. Los programas empleados simulan o interpretan los datos obtenidos de los archivos provenientes de los sistemas auditados, esta transferencia se realiza regularmente por medio de un servidor central.

Independientemente de la forma de implementación, la selección de la técnica de recolección asistida por tecnología requiere de una cuidadosa selección, esto debido a que algunas técnicas ofrecen nulos resultados cuando se aplican en sistemas para los cuales no fueron diseñados, un listado de ellas y sus características se presentan en la Tabla 3-13.

**Tabla 3-13 Técnicas asistidas por tecnología para recolección de información**

| Técnicas de Inspección asistidas por tecnología | Descripción |
|---|---|
| Comparación de programas | Esta técnica se emplea para efectuar una comparación de código (fuente, objeto o comandos de proceso) entre la versión de un programa catalogado en operación y la versión de un programa que ha sido analizado y debe permanecer en custodia para garantizar que no ha sido modificado en forma indebida. |
| Mapeo y rastreo de programas | Esta técnica emplea un software especializado que permite analizar los programas en ejecución, indicando el número de veces que cada línea de código es procesada y las condiciones de las variables de memoria que estuvieron presentes. También indica las líneas de código que no se utilizan durante el proceso. Esta técnica es especialmente útil para detectar problemas relacionados con virus de activación por tiempo. |
| Análisis de código de programas | Esta técnica se emplea para analizar los programas de una aplicación. El análisis puede efectuarse en forma manual (en cuyo caso sólo se podría analizar el código fuente de los programas) o utilizando software especial (para analizar tanto el código fuente como el código ejecutable). Esta técnica es útil para obtener un conocimiento del funcionamiento de un sistema o para verificar que el programa analizado incluye todos los procesos exclusivamente autorizados por el usuario de la aplicación. |

| Técnicas de Inspección asistidas por tecnología | Descripción |
|---|---|
| Datos de prueba | Esta técnica se emplea para verificar que los procedimientos de control incluidos en los programas de una aplicación funcionan correctamente. Los datos de prueba consisten en la preparación de una serie de transacciones que contienen tanto datos correctos como datos con errores predeterminados. Estas transacciones son alimentadas al sistema con la intención de que los filtros de control detecten los errores, al mismo tiempo que procesan adecuadamente la información que no contiene errores. Esta técnica es útil para verificar el correcto funcionamiento de procedimientos de control de detección de errores en un sistema. Sin embargo, no permite detectar rutinas diseñadas para reaccionar ante excepciones (cuando un programa reacciona ante una situación específica como una fecha determinada o un nombre propio en uno de los registros). |
| Datos de prueba integrados (Integrated Test Facilities) | Esta es una técnica similar a la anterior, con la diferencia de que en ésta se debe crear una entidad "falsa" dentro de los sistemas de información. Es decir, una sucursal, empresa o departamento inexistente, pero que sean procesados en forma conjunta con las transacciones reales de la compañía. (En los datos de prueba se trabaja normalmente con copias de programas y con respaldo de los archivos.) |
| Análisis de bitácoras | Existen varios tipos de bitácoras que pueden ser analizadas por el auditor, ya sea en forma manual o por medio de programas especializados, tales como bitácoras de uso del equipo, bitácoras de accesos no autorizados, bitácoras de uso de recursos, bitácoras de procesos ejecutados, bitácoras de fallas del equipo, etc. Las bitácoras contienen información histórica que permite al auditor efectuar análisis de utilización de recursos o detectar desviaciones a políticas o procedimientos de control establecidos. |
| Simulación paralela | Consiste en desarrollar programas o módulos que emulen a los programas de un sistema en producción. El objetivo es procesar los dos programas o módulos en forma paralela e identificar diferencias entre los resultados de ambos. En ocasiones, es factible que el propio programa de auditoría calcule las diferencias en forma automática. Sin embargo, en otras es necesario que esta técnica se complemente con técnicas de conciliación adicionales, mismas que pueden ser completamente manuales o auxiliadas por hojas electrónicas de cálculo. |
| Código embebido (embedded code) | Tal vez esta es la técnica más representativa de lo que debería ser una auditoría permanente, ya que se trata precisamente de dotar a los programas de un sistema de rutinas con técnicas de auditoría. Estos programas tienen la función de detectar anomalías o desviaciones en los parámetros o criterios aceptados por el sistema, ejecutándose de forma consistente, debido a que todas las veces que un programa sea ejecutado, también lo serán las rutinas de auditoría. |
| Análisis de datos | Esta técnica permite efectuar análisis de la información almacenada en archivos o bases de datos. Para este efecto se puede utilizar cualquier herramienta o lenguaje disponible para obtener información de una base de datos. El objetivo es identificar cualquier desviación a los parámetros permitidos para los registros de la base de datos, efectuar cálculos utilizando los datos contenidos en los archivos y generar reportes de auditoría. |
| Programas de utilerías (Utility Programs) | Esta, más que una técnica en sí, es la posibilidad de emplear los propios recursos de los sistemas a ser auditados con el fin de recabar información. |

*Fuente: López, 2001*

### 3.9.15 PLAN DE CONTINGENCIAS

Antes de abordar el tema de la elaboración del informe final, se revisara el concepto de "PLAN DE CONTINGENCIAS", elemento que no permitirán fundamentar las bases de un programa general de recuperación de desastres.

El plan de contingencia es una estrategia planificada constituida por un conjunto de recursos de respaldo, una organización de emergencia y procedimientos de ejecución encaminada a conseguir una restauración progresiva y ágil de los servicios de negocios afectados por una paralización total o parcial de la capacidad operativa de la empresa.

### 3.9.15.1 FASES DE UN PLAN DE CONTINGENCIAS

Las fases de desarrollo de un plan se presentan en la Tabla 3-14, las fases y objetivos de las mismas no son los únicas fases que existen, pero si los mas representativos para el tipo de auditoría que deseamos utilizar. Es posible que una empresa o institución cuente con sus propios manuales de procedimientos destinados a la elaboración de los planes de contingencia, en dado caso es necesario seguir de manera exacta los lineamientos establecidos en los manuales.

**Tabla 3-14 Fases del desarrollo de un plan de contingencias**

| FASE | DESCRIPCIÓN |
|---|---|
| FASE I<br>ANÁLISIS Y DISEÑO | Se estudia la problemática, las necesidades de recursos, las alternativas de respaldo, y se analiza el "costo-beneficio" de las mismas. Esta es la fase más importante pudiendo llegarse al final de la misma incluso a la conclusión de que no es viable o es muy costoso su seguimiento. Para el desarrollo de esta fase, se cuenta con dos metodológicas:<br>"RISK ANALYSIS" Basado en el estudio de los posibles riesgos desde el punto de vista de probabilidad de que los mismos sucedan.<br>"BUSSINES IMPACT" Se enfoca en el estudio del impacto (perdida económica o de imagen) que ocasiona la falta de algún recurso de los que soporta la actividad del negocio. |
| FASE II<br>DESARROLLO DEL PLAN | Esta fase y la tercera son similares en todas las metodologías. En ella se desarrolla la estrategia seleccionada, implantándose hasta el final todas las acciones previstas. Se definen las distintas organizaciones de emergencia y se desarrollan los procedimientos de actuación generando así la documentación del plan. |
| FASE III<br>PRUEBAS Y MANTENIMIENTO | En esta fase se definen las pruebas, sus características, y sus ciclos, y se realiza la primera prueba como comprobación de todo el trabajo realizado, así como mentalizar al personal implicado. Así mismo se define la estrategia del mantenimiento, las organizaciones destinadas a ello y la normatividad y procedimientos necesarios para llevarlo a cabo. |

*Fuente: López, 2001*

En la Fase I "Análisis y Diseño" (Ver Tabla 3-14), se presentan dos corrientes metodológicas para analizar los riesgos que pueden existir para nuestra aplicación o departamento. Tanto "RISK ANALYSIS" como "BUSSINES IMPACT", pueden ser empleados para evaluar los riesgos de cualquier empresa, no importando su tipo de actividad. Al respecto es necesario indicar que será la experiencia del auditor o la decisión de la empresa auditada optar por alguna de las dos metodologías.

En las Fases II y III del desarrollo de un plan de contingencias, se procesa toda la información recabada durante la Fase I, para dar forma a las acciones y medidas pertinentes, para dar paso a las pruebas de los procedimientos correctivos o de actuación propuestos. Estas labores hacen necesario disponer de una estrecha comunicación entre el departamento de auditoría y el resto de los departamentos afectados, para favorecer la implementación de dichos procedimientos.

## 3.9.16 CONTROL INTERNO

### 3.9.16.1 DEFINICIÓN DE CONTROL INTERNO

El "I.M.P.C." Define el Control Interno como: Es el sistema por el cual se da efecto a la administración de una entidad económica. En ese sentido, el término administración se emplea para designar el conjunto de actividades necesarias para lograr el objeto de la entidad económica. Abarca, por lo tanto, las actividades de dirección, financiamiento, promoción, distribución y consumo de una empresa; sus relaciones públicas y privadas y la vigilancia general sobre su patrimonio y sobre aquellos de quien depende su conservación y crecimiento.

## 3.9.17 CONTROL INTERNO INFORMÁTICO

El control interno informático inspecciona diariamente que todas las actividades del departamento de Sistemas de Información sean realizadas cumpliendo los procedimientos, estándares y normas fijados por la Dirección de la Organización o por la Dirección de Informática, así como los requerimientos legales.

## 3.9.18 OBJETIVOS

Los objetivos del control interno en general recaen en la búsqueda del mayor nivel de seguridad para resguardar los activos de una empresa, para lograrlo es necesario cumplir con los puntos que a continuación se listan:

- Controlar que todas las actividades se realizan cumpliendo los procedimientos y normas fijados, evaluar su bondad y asegurarse del cumplimiento de las normas legales.
- Asesorar sobre el conocimiento de las normas.
- Colaborar y apoyar el trabajo de Auditoría Informática, así como de las Auditorías externas al departamento.
- Definir, implantar y ejecutar mecanismos y controles para comprobar la realización de los objetivos del Departamento Informático.

## 3.9.19 CLASIFICACIÓN DE LOS CONTROLES INTERNOS

Debido al alcance tan amplio en sus funciones y objetivos de los Controles Informáticos se han clasificado en base a sus objetivos:

- Controles preventivos: para tratar de evitar el hecho, como un software de seguridad que impida los accesos no autorizados al sistema.
- Controles de guardia: cuando fallan los preventivos para tratar de conocer cuanto antes del evento. Por ejemplo, el registro de intentos de acceso no autorizados, el registro de la actividad diaria para detectar errores u omisiones, entre otros.

- Controles correctivos: facilitan la vuelta a la normalidad cuando se han producido incidencias. Por ejemplo, la recuperación de un archivos dañado a partir de las copias de seguridad.

Cada categoría pude y debe interactuar entre si, con la finalidad de garantizar un nivel alto de seguridad en el departamento al cual controlan. Para la implementación de un sistema de controles internos informáticos, se deberá previamente definir las actividades relacionadas con; la gestión de sistemas de información, la administración de sistemas, la seguridad y el control o gestión de cambios. Actividades que se listan a continuación:

- Gestión de sistemas de información: Políticas, pautas y normas técnicas que sirvan de base para el diseño y la implantación de los sistemas de información y de los controles correspondientes
- Administración de sistemas: Controles sobre la actividad de los centros de datos y otras funciones de apoyo al sistema, incluyendo la admón. de las redes.
- Seguridad: Incluye las tres clases de controles fundamentalmente integrados en el software del sistema, integridad del sistema, confidencialidad (control de acceso) y disponibilidad.
- Gestión de cambio: Separación de las pruebas y la producción a nivel de software y controles de procedimientos para la migración de programas software aprobados y probados.

### 3.9.20 CLASIFICACIÓN DE LOS CONTROLES INTERNOS A NIVEL ORGANIZATIVO

Debido a que el control interno puede enfocarse en las diversas actividades del departamento de sistemas, es necesario no olvidar que el control interno es un instrumento que deberá estar ligado a la organización de la empresa, motivo por el cual se agrupan los tipos de control en función de labores departamentales, de esta forma nuestra clasificación es la siguiente:

- Controles generales organizativos
- Controles de desarrollo, adquisición y mantenimiento de sistemas de información
- Controles de explotación de sistemas de información:
- Controles de aplicación
- Controles específicos de ciertas tecnologías

Cada uno de los cuales será detallado a continuación.

#### 3.9.20.1.1 CONTROLES GENERALES ORGANIZATIVOS

Los controles generales organizativos se encuentran inmersos en todas las actividades del departamento de informática, los controles generados en este nivel deberán servir de base a la planificación, control y evaluación por la dirección de las actividades del departamento de informática. En este nivel se desarrollan estándares que regularan la adquisición de recursos, el diseño, desarrollo, modificación y explotación de los sistemas, actividades que como se expone en las secciones 3.9.20.2 y 3.9.20.3 disponen de sus propios controles, los

cuales sirven para supervisas aspectos mas detallados de las labores realizadas por este departamento.

Los Controles Generales Organizativos disponen de procedimientos que describen la forma y las responsabilidades ejecutorias para regular las relaciones entre el departamento de informática y los departamentos existentes en el organigrama. La organización de las funciones y posición del departamento de informática se realiza basado en los controles generales, motivo por el cual se puede garantizar la correcta funcionalidad entre los departamentos a los cuales sirve y de los cuales depende. En cuanto a la localización del departamento de informática dentro del organigrama, se busca localizarlo en un nivel suficientemente superior de la estructura organizativa para asegurar su independencia de los departamentos a los cuales sirve y dependencia suficiente para asegurar que los niveles de dirección revisen los informes de control y resuelvan las excepciones que ocurran.

Esta estructura de control lleva a cabo una planificación integral del departamento de informática, basados en los planes estratégicos productos de las diversas unidades del propio departamento (Ver Tabla 3-15).

**Tabla 3-15 Elementos que conforman los programas de planificación del control general**

| Planificación | Descripción |
|---|---|
| Plan estratégico de información | Es realizado por los órganos de alta dirección de la empresa donde se definen los procesos corporativos y se considera el uso de diversas tecnologías de información así como las amenazas y oportunidades de su uso o de su ausencia. |
| Plan informático | Es realizado por el departamento de informática, determina lo caminos precisos para cubrir las necesidades de la empresa plasmándolas en proyectos informáticos. |
| Plan general de seguridad (física y lógica) | Es realizado para presentar acciones, esquemas y recomendaciones encaminadas a garantizar la confidencialidad, integridad y disponibilidad de la información. |
| Plan de emergencias ante desastres | Es realizado con el fin de contemplar todos los posibles eventos que representen un inconveniente para la correcta o total funcionalidad del sistema informático. El objetivo de un plan de emergencia es el de proveer acciones que garanticen la disponibilidad de los sistemas lo antes posible después de ocurrido el desastre. |

*Fuente: López, 2001*

### 3.9.20.1.2 CONTROLES DE DESARROLLO, ADQUISICIÓN Y MANTENIMIENTO DE SISTEMAS DE INFORMACIÓN

Los controles de desarrollo, adquisición y mantenimiento de sistemas de información, tienen por objetivo vigilar y establecer los procedimientos establecidos para el desarrollo, adquisición y mantenimiento de todos los programas de computadoras de uso general, propiedad de la empresa. En cuestiones de asuntos legales es deber del control de desarrollo, ofrecer orientación y solucionar problemas relacionados con derechos de uso, licenciamiento, entre otros.

### 3.9.20.2 CONTROLES DE EXPLOTACIÓN DE SISTEMAS DE INFORMACIÓN

El establecimiento de controles a los recursos disponibles, asegurara que los datos se tratan de forma congruente y exacta, que el contenido de los sistemas solo será modificado mediante la autorización adecuada. Trabajando en conjunto con otros controles internos se generan los planes de respaldo y disponibilidad de la información. Algunas de las acciones de este tipo de control son:

- Planificación y gestión de recursos
- Establecimiento de mecanismos dirigidos a incrementar el uso eficiente de los recursos disponibles en la infraestructura informática.
- Diseño y ejecución de procedimiento de selección del software del sistema, de instalación, de mantenimiento, de seguridad y control de cambios. Tarea que realiza en conjunto con los controles de desarrollo, adquisición y mantenimiento de sistemas.
- Inspección y corrección de los sistemas y esquemas de seguridad física y lógica

### 3.9.20.3 CONTROLES DE APLICACIÓN

Cada aplicación debe llevar controles incorporados para garantizar la entrada, actualización, validez y mantenimiento completos y exactos de los datos. Las acciones más importantes en el control de los datos son:

- Controles de entrada de datos: Procedimientos de conversión y de entrada, validación y corrección de datos.
- Controles de tratamientos de datos para asegurar que no se dan de alta, modifican o borran datos no autorizados para garantizar la integridad de los mismos mediante procesos no autorizados.
- Controles de salida de datos: Sobre el cuadre y reconciliación de salidas, procedimientos de distribución de salidas, de gestión de errores en las salidas, etc.

#### 3.9.20.3.1 CONTROLES ESPECÍFICOS DE CIERTAS TECNOLOGÍAS

Estos controles tienen por objetivo regular la adquisición, explotación y administración de elementos de la infraestructura informática considerados indispensables para el funcionamiento de la organización. En general se enfoca en los recursos físicos como pueden ser el uso y distribución de los recursos de la red de telecomunicaciones, el almacenamiento disponible en los servidores de archivos, entre otros recursos disponibles. En forma general los controles que se aplican son:

- Controles de sistemas de gestión de bases de datos
- Controles en informática distribuida y redes
- Controles sobre computadoras personales y redes de área local

### 3.9.21 METODOLOGÍA DE OBTENCIÓN DE PROCEDIMIENTOS DE CONTROL

Identificados los tipos de control requeridos, es necesario generar los controles y llevar un sistema de verificación de pruebas e implantación de controles. El procedimiento de obtención e implantación de controles, se compone de tres fases principales que son:

- Definición de objetivos de control.
- Definición de los controles
- Implantación de los controles

Cada uno de ellas se describe adecuadamente en la Tabla 3-16.

**Tabla 3-16 Metodología de obtención de procedimientos de control**

| Fases | Tareas |
|---|---|
| Fase1. Definición de objetivos de control. | Análisis de la empresa. Se estudian los procesos, organigramas y funciones.<br>Recopilación de estándares. Se estudian todas las fuentes de información necesarias para conseguir definir en la siguiente fase los objetivos de control a cumplir (por ejemplo, ISO[3], CISA[4], entre otros.)<br>Definición de los objetivos de control |
| Fase 2. Definición de los controles | Definición de los controles. Con los objetivos de control definidos, analizamos los procesos y vamos definiendo los distintos controles que se necesiten.<br>Definición de necesidades tecnológicas (hardware y herramientas de control).<br>Definición de los procedimientos de control. Se desarrollan los distintos procedimientos que se generan en las áreas usuarias, informática, control informática, control informático y control no informático.<br>Definición de las necesidades de recursos humanos. |
| Fase 3 Implantación de los controles | Una vez definidos los controles, las herramientas de control y los recursos humanos necesarios, no resta más que implantarlos en forma de acciones específicas. |

*Fuente: López, 2001*

Terminado el proceso de implantación de acciones habrá que documentar los procedimientos nuevos y revisar los afectados de cambio. Los procedimientos resultantes serán:

- Procedimientos de distintas áreas de usuarios de la informática, mejorados.
- Procedimientos de áreas informáticas, mejorados.
- Procedimiento de control dual entre control interno informático y el área de informática, los usuarios informáticos y el área de control no informático.

---

[3] **ISO**: (International Organization for Standardization, Organización Internacional para los estándares )

[4] **CISA**: (Certified Information Systems Auditor, Auditoría Certificada en Sistemas de Información.)

### 3.9.21.1 HERRAMIENTAS DE CONTROL

Las herramientas de control son elementos de software que por sus características funcionales permiten vertebrar un control de manera eficiente y automatizada. Las clases de herramientas de control de software de uso más común son: Seguridad lógica del sistema, seguridad lógica complementaria al sistema, seguridad lógica para entornos distribuidos, control de proyectos, entre otros.

### 3.9.21.2 SELECCIÓN DE LAS HERRAMIENTAS DE CONTROL

Debido a que las herramientas de control son instrumentos necesarios para evaluar el funcionamiento de los controles establecidos, se hace necesario contar con una metodología de selección, siguiendo una o varias metodologías (Ver Tabla 3-17) es posible seleccionar adecuadamente el Software que nos ayude a monitorear eficientemente el desempeño de los controles que han sido implementados.

**Tabla 3-17 Metodologías de selección de herramientas de control**

| Metodología | Descripción |
|---|---|
| ANÁLISIS DE PLATAFORMAS | Se trata de inventariar las múltiples plataformas actuales y futuras que nos servirán para saber que productos del mercado no pueden ser validos, tanto los productos actuales como los futuros planes que tengan los fabricantes. |
| CATALOGO DE REQUERIMIENTOS PREVIOS DE IMPLANTACIÓN | Se inventaría lo que no se va a conseguir (limitaciones), así como los necesarios para la implantación, inventariado como acciones y proyectos y su duración para seguimiento y desarrollo. |
| ANÁLISIS DE APLICACIONES | Se trata de inventariar las necesidades de desarrollar interfaces con los distintos sistemas de seguridad de las aplicaciones y base de datos. |
| ADMINISTRACIÓN DE LA SEGURIDAD | Se analizarán, de las distintas operaciones del mercado, las características de cada producto. |
| SINGLE SING ON. | Este concepto podemos definirlo como: "Que es necesario para un usuario, solamente una contraseña y un "identificador", para acceder y usar su información y recursos, de todos los sistemas como si de un solo entorno se tratara". |
| SEGURIDAD | Trata aspectos de seguridad clásicos del propio producto, como que el administrador no vea la contraseña de los usuarios, que se requiera de una combinación Nombre de usuario – Contraseña, el administrador pueda suspender un usuario determinado, restringir el acceso a un recurso local por parte de un usuario, entre otras funciones. |
| FACILIDAD DE USO Y GENERACIÓN DE REPORTES | Se valora la interfaz de usuario y la calidad de la misma. Se evalúa el nivel de generación de reportes para incluso los administradores y auditores. Todo registro debe tener garantizada su integridad incluso para los administradores, no pudiendo desactivarse a voluntad, dado que quien quiera hacer algo "no permitido", lo primero que hará es asegurarse de que no quede constancia del hecho. |
| INVENTARIO DE FUNCIONALIDADES Y PROPIETARIOS | Se trata de todo el esquema de funcionalidades de la seguridad lógica actual. Es el momento de crear unas jerarquías de estándares a cumplir y tratar de definir en ese momento los controles que se deberían de tener, ya desea de usuarios de las aplicaciones como los usuarios de los sistemas y el uso de las herramientas.<br>Es importante inventariar la situación de la administración de la seguridad lógica en los distintos entornos de los distintos sistemas como de las distintas administraciones de seguridad y el control de reportes.<br>Este inventario nos servirá para hacer un análisis de mejoras y pérdidas de limitaciones en los nuevos escenarios con el software de control de los entornos distribuidos, según convenga para elegir el mejor "costo – beneficio". |

*Fuente: López "2001"*

### 3.9.22 EL INFORME DE AUDITORÍA

En esta sección se presentan los elementos y estructura necesaria para elaborar el informe, producto final de una auditoría interna. El informe comprende aspectos como lo son: las normas de auditoría, las evidencias en auditoría, las irregularidades, los papeles de trabajo o documentación, para, finalmente, encarar el informe. La responsabilidad del auditor se centra en planificar, llevar acabo y evaluar su trabajo para obtener una expectativa razonable de su detección.

#### 3.9.22.1 LA DOCUMENTACIÓN

En auditoría se le conoce como papeles de trabajo a la "totalidad de los documentos preparados o recibidos por el auditor, de manera que, en conjunto constituye un compendio de la información utilizada y de las pruebas efectuadas en la ejecución de su trabajo, junto con las decisiones que a debido tomar para llegar a formarse su opinión". Además se incluirán:

El contrato cliente/auditor informático y/o la carta propuesta del auditor informático.

- Las declaraciones de la dirección.
- Los contratos, o equivalentes, que afecten al sistema de información así como el informe de la asesoría jurídica del cliente sobre sus asuntos actuales y previsibles.
- El informe sobre terceros vinculados.
- Conocimiento de la actividad del cliente.

#### 3.9.22.2 EL INFORME

El informe deberá ser claro, adecuado, suficiente y comprensible. Una utilización apropiada del lenguaje informático resulta recomendable.

Los puntos esenciales, genéricos y mínimos del informe de Auditoría Informática son doce:

- **Identificación del informe.** El título del informe deberá identificarse con objeto de distinguirlo de otros informes.
- **Identificación del cliente.** Deberá identificarse a los destinatarios y a las personas que efectúan el encargo.
- **Identificación de la entidad auditada.** Identificación de la entidad objeto de la Auditoría informática.
- **Objetivos de la Auditoría informática.** Declaración de los objetivos de la Auditoría informática para identificar su propósito, señalando los objetivos incumplidos.
- **Normativa aplicada y excepciones.** Identificación de las normas legales y profesionales utilizadas, así como las excepciones significativas de uso y el posible impacto en los resultados de la Auditoría.
- **Alcance de la Auditoría.** Concretar la naturaleza y extensión del trabajo realizado: área organizativa, período de Auditoría, sistema de información..., señalando limitaciones alcance y restricciones del auditado.

- **Conclusiones.** Informe corto de opinión. Lógicamente, se ha llegado a los resultados y, sobre todo, a la esencia del dictamen, la opinión y los párrafos de salvedades y énfasis, si procede. El informe debe contener algunos de los siguientes tipos de opinión: Favorable o sin salvedades, con salvedades, desfavorable o adversa, y denegado.
    - Opinión favorable. Debe manifestarse de forma clara y precisa, y es el resultado de un trabajo realizado sin limitaciones de alcance y sin incertidumbre, de acuerdo con la normativa legal y profesional.
    - Opinión con salvedades. Podrán ser estas, según las circunstancias, las siguientes:
        - Limitaciones al alcance del trabajo realizado; esto es, restricciones por parte del auditado, entre otros eventos.
        - Incertidumbres cuyo resultado no permita una previsión razonable.
        - Irregularidades significativas.
        - Incumplimiento de la normativa legal y profesional.
- Opinión Desfavorable. Es aplicable en el caso de:
    - Identificación de irregularidades.
    - Incumplimiento la normativa legal y profesional, que afecten significativamente a los objetivos de la Auditoría informática estipulados, incluso con incertidumbres; todo ello en la evaluación de conjunto y reseñando detalladamente las razones correspondientes.
- Opinión Denegada. La denegación de opinión puede tener su origen en:
    - Las limitaciones al alcance de Auditoría.
    - Incertidumbres significativas de un modo tal que impidan al auditor formarse una opinión.
    - Irregularidades.
    - Incumplimiento de normativa legal y profesional.
- **Resumen.** El siempre difícil tema de la opinión, estrella del informe Auditoría informática, joven como informática y más todavía como Auditoría informática; por tanto, puede decirse que más que cambiante, mutante. Debido a ello, y además con la normativa legal y profesional desacompasadas, la ética se convierte casi en la única fuente de orientación para reducir el desfase entre las expectativas del usuario en general y el informe de los auditores
- **Resultados: informe largo y otros informes.** Parece ser que, de acuerdo con la teoría de los ciclos, el informe largo coloca al informe corto en su sitio, o sea, como resumen del informe largo. Los usuarios deben saber la transparencia y claridad como valor añadido.
- **Informes previos.** No es una práctica recomendable, pero si usual en algunas partes ya que, el informe de Auditoría es un informe de conjunto. En el caso de la detección de irregularidades significativas como fraudes, requieren de una actuación inmediata según la normativa legal y profesional.
- **Fecha del informe.** El tiempo no es neutral, la fecha del Informe es importante, no solo para cuantificar los honorarios del equipo auditor, sino para conocer la magnitud del trabajo y sus implicaciones. Conviene precisar las fechas de inicio y conclusión del trabajo de campo, incluso la del cierre del ejercicio, si es que está

realizando un informe de Auditoría informático como herramienta de apoyo de cuentas.

- **Identificación y firma del auditor**. Este aspecto formal del informe es esencial tanto si es individual o parte de una sociedad de Auditoría, que deberá corresponder a un socio o socios legalmente constituidos.
- **Distribución del informe.** Deberá definirse a quien o quienes podrán hacer uso del informe, así como los usos concretos que tendrá, pues los honorarios deberán guardar relación con la responsabilidad civil.

### 3.9.23 ELEMENTOS BÁSICOS EN AUDITORÍAS DE SISTEMAS

#### 3.9.23.1 AUDITORÍA DE LA DIRECCIÓN DE SISTEMAS.

La dirección de Sistemas no puede quedar fuera de la Auditoría de Sistemas de Información, ya que éste departamento está totalmente influenciado por ésta. En la dirección de informática podemos incluir las siguientes actividades:

- PLANIFICACIÓN
- ORGANIZACIÓN Y COORDINACIÓN
- CONTROL

#### 3.9.23.2 PLANIFICACIÓN

Se trata de prever la utilización de las tecnologías de la información en la empresa. Existen varios tipos de planes informáticos; el principal y el origen de los demás es el "Plan Estratégico de Sistemas de Información". El cual se puede entender como: Es el marco básico de actuación de los Sistemas, de Información en la empresa. Debe asegurar el alineamiento de los mismos con los objetivos de la propia empresa. La dirección de Informática debe de ser el permanente impulsor de una planificación de Sistemas de Información adecuada y a tiempo. La vigencia de un plan estratégico es de 3 a 5 años, todo depende del entorno en el que se mueve la empresa.

El auditor deberá examinar el proceso de planificación de sistemas de información y evaluar si razonablemente se cumplen los objetivos para el mismo. Entre otros aspectos deberá evaluar si:

- Durante el proceso de planificación se presta adecuada atención al plan estratégico de la empresa, si se tienen en cuenta aspectos como cambios organizativos, entorno legislativo, evolución tecnológica, organización informática, recursos, etc. y sus impactos están adecuadamente recogidos en el Plan Estratégico de Sistemas de Información.
- Las tareas y actividades presentes en el Plan tienen la correspondiente y adecuada asignación de recursos para poder llevarlas a cabo.

### 3.9.23.3 ORGANIZACIÓN Y COORDINACIÓN

El proceso de organizar sirve para estructurar los recursos, los flujos de información y los controles que permitan alcanzar los objetivos marcados durante la planificación.

**Tabla 3-18 Departamentos y objetivos a ser evaluados durante una auditoria de dirección de sistemas**

| Organización | Elementos a ser evaluados |
|---|---|
| Comité de Informática | Es el primer lugar de encuentro dentro de la empresa de los informáticos y sus usuarios: es el lugar en el que se debaten los grandes asuntos de la informática que afectan a toda la empresa y permite a los usuarios conocer las necesidades del conjunto de la organización. El Comité deberá estar formado por pocas personas y presidido por el director más experimentado, dentro de la empresa, responsable en último término de las tecnologías de la información. El director de informática debería actuar como secretario del Comité y las grandes áreas usuarias deberán estar representadas al nivel de sus directores. Asimismo el director de Auditoría de Sistemas de Información debería ser miembro del Comité de Informática. |
| Posición del Departamento de Informática dentro de la Empresa | El segundo aspecto importante a tener en cuenta a la hora de evaluar el papel de la informática en la empresa es la ubicación del Departamento. El Departamento de Informática debería estar suficientemente alto en la jerarquía y contar con masa crítica suficiente para disponer de autoridad e independencia frente a los departamentos usuarios. |
| Descripción de Funciones y Responsabilidades del Departamento de Informática | Es necesario que se tengan las funciones descritas y sus responsabilidades claramente delimitadas y documentadas, todo el personal perteneciente a este departamento debe conocer sus funciones y responsabilidades en relación con los sistemas de información. |
| Estándares de funcionamiento y procedimientos | Deben existir estándares que gobiernen la actividad del departamento de informática y sus relaciones con los departamentos usuarios. De particular importancia son los aspectos relacionados con la adquisición de equipos o material para el departamento, con el diseño y el desarrollo/modificación de sistemas de información y con la producción o explotación. Deben existir descripciones documentadas de los puestos de trabajo dentro de Informática delimitando claramente la autoridad y responsabilidad en cada caso. Las descripciones deberá incluir los conocimientos técnicos y/o experiencia necesarios para cada puesto de trabajo. |
| Gestión de Recurso Humano | La calidad del recurso humano influye directamente en la calidad de los sistemas de información producidos, mantenidos y operados por el departamento de informático. Seleccionarlos, mantenerlos y motivarlos adecuadamente puede ser crucial para la buena marcha de la informática y su papel en la empresa. |
| Comunicación | Es necesario que exista una comunicación efectiva y eficiente entre la Dirección de Informática y el resto del personal del Departamento. Entre los aspectos que es importante comunicar se encuentran: actitud positiva hacia los controles, integridad, ética., cumplimiento de la normativa interna, compromiso con la calidad, etc. |
| Gestión Económica | En necesaria la existencia de un presupuesto anual, de un proceso para la elaboración del mismos, que incluya consideraciones de usuarios, y que este de acuerdo con las política, procedimientos, plan estratégico, y operativo del departamento. Así como la existencia de procedimientos para adquisición de bienes y servicios descritos en el plan operativo, estos deben ser documentados y alineados con los procesos de compra de la empresa. |
| Seguros | El auditor deberá estudiar las pólizas de seguros y evaluar su cobertura. |

*Fuente: López, 2001*

### 3.9.23.4 CONTROL

Se ha de vigilar el desarrollo del plan estratégico y operativo, de los proyectos que se desarrollan, la ejecución del presupuesto, entre otros.

La dirección debe controlar aspectos como normativa laboral y sindical, protección de datos personales, propiedad intelectual del software, requisitos definidos en la cobertura de seguros, y demás aspectos. El auditor deberá evaluará si la normativa aplicable se cumple.

### 3.9.24 AUDITORÍA DE APLICACIONES

Una aplicación informática o sistema de información habitualmente persigue como finalidad:

- Registrar fielmente la información considerada de interés en torno a las operaciones llevadas a cabo por una determinada organización.
- Permitir la realización de cuantos procesos de cálculo y edición sean necesarios a partir de la información registrada, permitiendo por lo tanto almacenar automáticamente más información que la de partida, aunque siempre basada en la información original.
- Facilitar, a quienes lo precisen, respuesta a consultas de todo tipo sobre la información almacenada, diseñadas en contenido y forma para dar cobertura a las necesidades más comunes constatadas.
- Generar informes que sirvan de ayuda para cualquier finalidad de interés en la organización, presentando la información adecuada: se aplican "según convenga" criterios de selección, ordenación, recuento y totalización por agrupamientos, cálculos de todo tipo.

Identificar fallas sutiles que representen el principio de algunas amenazas al normal cumplimiento de la finalidad de nuestra aplicación:

- La posibilidad de fallo en cualquiera de los elementos que intervienen en el proceso informático: software perteneciente a diferentes firmas, Servidores y dispositivos periféricos, transmisión de datos (servidores, bancos de módems, sistemas de telecomunicaciones, entre otros.) constituye otra fuente de posibles riesgos.
- La conexión cada vez más generalizada de las empresas a entornos abiertos como la Internet multiplica los riesgos que amenazan la confidencialidad e integridad de la información de nuestros sistemas.

Para cada una de ellas se habrán debido estudiar las posibles medidas tendentes a eliminar los riesgos que entrañan o, cuando menos, a reducir la probabilidad de su materialización hasta niveles razonablemente asimilables, siempre teniendo en cuenta el costo de tales medidas.

En la Auditoría de una aplicación se trata de realizar una revisión de la eficacia del funcionamiento de los controles diseñados para cada uno de los pasos de la misma frente a los riesgos que tratan de eliminar o minimizar, como medios para asegurar la fiabilidad (totalidad y exactitud), seguridad, disponibilidad y confidencialidad de la información gestionada por la aplicación.

### 3.9.25 AUDITORÍA DEL DESARROLLO

Para delimitar el ámbito de la Auditoría del desarrollo, se entenderá que el desarrollo incluye todo el ciclo de vida del software excepto la explotación, el mantenimiento y la retirada de servicio de las aplicaciones cuando ésta tenga lugar.

Existen una serie de circunstancias por las cuales es especial la Auditoría al departamento de desarrollo:

- El gasto destinado a software es cada vez superior al que se dedica a hardware.
- Los avances en tecnologías de las computadoras han hecho que actualmente el desafío más importante y el principal factor de éxito de la informática sea la mejora de la calidad del software.
- La "Crisis del Software" (problemas asociados con el desarrollo y mantenimiento del software).
- El software como producto es muy difícil de validar. Un mayor control en el proceso de desarrollo incrementa la calidad del mismo y disminuye los costos de mantenimiento.
- El índice de fracasos en proyectos de desarrollo es demasiado alto, lo cual denota la inexistencia o mal funcionamiento de los controles en este proceso.

### 3.9.25.1 PLANTEAMIENTO Y METODOLOGÍA

Para tratar la Auditoría del área de desarrollo es necesario conocer las funciones que tradicionalmente se asignan en ésta área:

- Planificación del área y participación, en la medida que corresponda, en la elaboración del plan estratégico de informática
- Desarrollo de nuevos sistemas. Esta es la función principal y la que da sentido al área de desarrollo.
- Estudio de nuevos lenguajes, técnicas, metodologías, estándares, herramientas, etc. relacionados con el desarrollo y adopción de los mismos cuando se considere oportuno para mantener un nivel de vigencia adecuado a la tecnología del momento.
- Establecimiento de un plan de formación para el personal adscrito al área
- Establecimiento de normas y controles para todas las actividades que se realizan en el área y comprobación de su observancia.

# CAPÍTULO 4 DESARROLLO DE INTERFACES Y SERVICIOS

Las interfaces y servicios generados para la solución Web, son el marco que expondrá toda la funcionalidad diseñada en el modelo de caso de usos, sin un desarrollo adecuado y controlado nada del trabajo realizado se verá plasmado en las pantallas de los usuarios, lo cual representa un fracaso al no poder relacionar toda la funcionalidad existente en las capas de: negocios, acceso a datos y almacén de datos.

Al proponer el empleo de una solución basada en Web, la visión de nuestro cliente es proveer a sus usuarios de una aplicación de fácil acceso, alta disponibilidad, acceso controlado y ejecución sencilla. Las aplicaciones de escritorio (aplicaciones residentes en el equipo cliente), ofrecen mayor velocidad de ejecución y tiempos de respuesta más bajos, pero no satisfacen los requerimientos de alta disponibilidad, debido a que requieren ser instaladas en la computadora que en ese momento desee emplear el usuario. En muchos casos las aplicaciones de escritorio cuentan con una extensa variedad de menús y opciones, las cuales muchas veces presentan problemas para los usuarios inexpertos, situación que en las aplicaciones basadas en Web es casi inexistente debido a la sencillez de los controles e interfaces empleadas.

En el presente capitulo se dedica una sección a la introducción a un servicio Web desarrollado para incrementar los servicios que este proyecto ofrece. La naturaleza de la información que maneja el proyecto se encuentra sujeta a un contrato de confidencialidad entre el Gobierna de la Ciudad de México y el Instituto de Ingeniería, por lo que algunos aspectos en cuanto al manejo de ellos serán comentados de forma general.

# 4.1 SELECCIÓN DE HERRAMIENTAS

## 4.1.1 PLATAFORMA DE DESARROLLO. NET FRAMEWORK

Los antecedentes del por qué desarrollar en la plataforma de .Net, es debido a que nos proporciona un marco de trabajo escalable para las necesidades de las aplicaciones desarrolladas, permitiendo alojar las aplicaciones que se encuentran ya realizadas y poder agregar complementos funcionales que permiten de forma sencilla integrarlos sin necesidad de reestructurar la aplicación completa o tener que realizar una inversión mayor por cuestiones de recursos de computo, software o personal. Un mayor detalle de las cualidades de esta plataforma se puede hacer referencia al capitulo 2 en la sección de requisitos.

El uso del .Net Framework, nos proporciona la plataforma necesaria para la implementación de la solución Web propuesta, debido a la necesidad de implementar los esquemas de trabajo, seguridad e integración con los elementos que componen las solución Web, la plataforma cuenta con elementos que facilitarán el desarrollo y compatibilidad con la plataforma de producción.

## 4.1.2 CARACTERÍSTICAS DEL .NET FRAMEWORK

El Framework a emplearse tiene como principal objetivo proveer de un marco de trabajo sólido, orientado a objetos y que permita a los programadores abstraerse de tareas rutinarias con el empleo de las librerías contenidas en el mismo. En la Figura 4-1 se aprecia la integración nativa que ofrece esta plataforma. Para nuestro proyecto esta integración abarca las clases de: acceso a datos ("ADO.NET"), presentación ("ASP.NET"), reglas de negocio ("CLS") y ambiente de ejecución ("IIS").

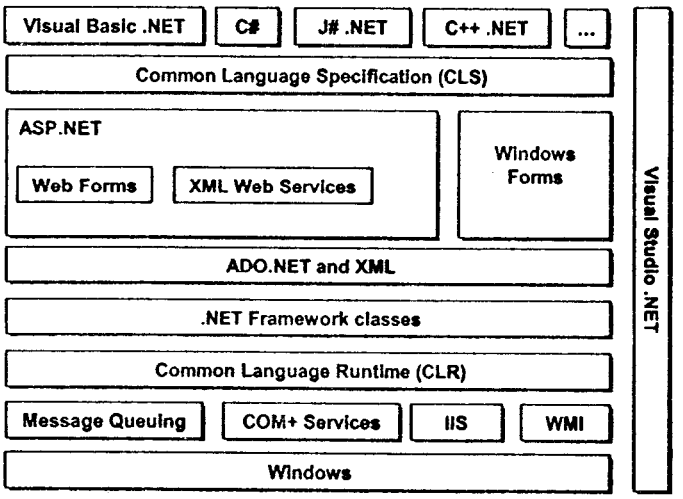



**Figura 4-1 Arquitectura del .Net Framework**

Para concentrar los beneficios considerados en la selección de esta plataforma, a continuación listaremos algunas características:

- Alto desempeño: Cuenta con una avanzada compilación y técnicas de caché, integración nativa al contenedor Web y con el hardware de producción, las aplicaciones de servidor son más rápidas que las creadas en PHP y JSP. Para el caso de las aplicaciones creadas en PHP, por cada petición es necesario interpretar el contenido del objeto requerido. Para las aplicaciones J2EE, el contenedor Web tiene la responsabilidad de crear una instancia del servlet y de invocar al método init. Si un cliente ha enviado una petición al contenedor Web, entonces, esa petición se pasa al método servicio del servlet y se envía una respuesta de vuelta al contenedor Web, finalmente, cuando el servlet haya finalizado su propósito, el contenedor Web invoca al método destroy.
- Seguridad: La tecnología de seguridad de acceso al código del Framework fue diseñada para los ambientes actuales de Internet. Cuenta con clases para recolectar evidencia acerca de dónde se origina una aplicación, quién la creó, cuál es su firma digital y qué está tratando de hacer la aplicación. El ambiente de ejecución puede combinar esa evidencia con políticas de seguridad para decidir si una aplicación puede acceder a determinado recurso.
- Integración con sistemas existentes: La tecnología incluida en el Framework para la operación con objetos COM genera una envoltura en sus componentes COM existentes, permitiendo programar para ellos como si hubieran estado diseñados originalmente para esta plataforma. Las aplicaciones creadas usando el Framework pueden conectarse con sistemas y aplicaciones existentes, sin importar su plataforma base.
- Soporte nativo para servicios Web XML: El Framework incluye soporte para servicios Web XML, un modelo para computación distribuida en múltiples ambientes basados en protocolos estándar como XML, SOAP y HTTP.
- Acceso flexible a datos: La tecnología del Framework para acceso a datos es ADO.NET, la cual presenta opciones de trabajar con conexión a base datos o en modo desconectado por medio de caché de datos basado en XML. Al igual que en J2EE, el objetivo es liberar recursos en la base de datos. Con el empleo de la plataforma seleccionada se presenta la posibilidad de transmitir información estructurada a través de redes e intraredes de manera transparente.

### 4.1.3 LENGUAJES DE PROGRAMACIÓN

#### 4.1.3.1 C#

El uso de C# para el desarrollo de la solución Web se basa en la necesidad de integrar de forma optimizada los procesos y rutinas críticas en el manejo de los datos, controles de acceso, control de procesos y elementos que componen la aplicación como son: interfaces de usuario, interfaces para la interoperabilidad entre controles de usuario, manipulación de objetos ADO como son las clases de conexión a bases de datos, ejecución y recuperación de consultas, clases de encriptación y validación de credenciales.

Entre las ventajas de la utilización de C#, se encuentra la utilización del sistema de tipos comunes lo cual permite aprovechar el código escrito en otros lenguajes compatibles, a la vez que pueden realizar llamadas a procedimientos remotos.

#### 4.1.3.1.1 CARACTERÍSTICAS

- Lenguaje basado en la sintaxis de C++, característica que comparte Java, facilitando la migración de código ya existente en Java.
- Posibilidad de uso de código "no seguro" para activar API de Windows nativas, COM, ActiveX, entre otras opciones. Característica que al igual que en Java debe ser controlada para evitar excepciones que involucren fallas.
- Tipos de estructuras: En Java clases e interfaces. En C# clases, interfaces, struct y enum.
- Elementos que pueden definirse dentro de una clase: En Java; atributos, métodos y clases internas. En .C# atributos, métodos, clases internas, propiedades, eventos y delegates.
- Niveles de encapsulamiento: En Java public, private, protected y visibilidad de paquete, este último se asume cuando se omite. En C# public, private, internal, protected y la combinación de estos dos últimos. En caso de omisión se asume private.
- Herencia: En Java no se permite la herencia múltiple, se puede simular a través del uso de interfaces. Por omisión se hereda de Object. En C# no se permite la herencia múltiple, se puede simular a través del uso de interfaces. Por omisión se hereda de Object.
- Polimorfismo: En Java, se permite que una clase sobrecargue o sobrescriba métodos definidos por su clase padre, a menos que la clase padre lo impida mediante la palabra reservada final, en el encabezado del método. Si un objeto de una clase hija es referenciado a su clase padre, su comportamiento, al invocar un método sobrescrito, será el que definió la clase a la cual él pertenece. En C# se permite que una clase sobrecargue o sobrescriba métodos definidos por su clase padre, a menos que la clase padre lo impida empleando la palabra reservada sealed, en el encabezado del método. Si un objeto de una clase hija es referenciado a su clase padre, su comportamiento, al invocar un método sobrescrito, dependerá de los permisos establecidos por la clase padre, y de la decisión tomada por quien definió la clase. Por omisión se comportará como lo definió la clase padre.
- Sobrecarga de operadores para una clase: En Java no permite la sobrecarga de ninguno de los operadores básicos. En C# se permite la sobrecarga de algunos de los operadores básicos: Unitarios: +, -, !, ~, ++, --, true, false; Binarios: +, -, *, /, %, &, |, ^, <<, >>, ==, != , >, <, >=, <=.

### 4.1.4 MICROSOFT, VISUAL STUDIO .NET

La suite de Desarrollo Visual Studio .Net es un conjunto de herramientas entre las cuales están el acceso a bases de datos, servicios y recursos locales y remotos (colas de mensajes, registro de sucesos, registro del sistema entre otros). Este grupo de herramientas cuenta con las siguientes características:

- Soporte a los lenguajes de desarrollo (C#, Visual Basic, J#, entre otros)
- Generación de Código de rutinas comunes de forma automática
- Desarrollo gráfico para aplicaciones visuales
- Soporte para componentes COM, .Net, entre otros
- Soporte para multimedia
- Integración con sistemas de almacenamiento de bases de datos (SQL; MySQL, Oracle)
- Compatible con formatos HTML, HTM, XML, ASP, ASPX entre otros formatos para páginas Web

Esta suite de desarrollo cuenta con una gran infraestructura de recursos entre ellas control de código, herramientas auxiliares, componentes de terceros, ayuda, asesoría, libros en línea, ejemplos, etc. Permitiendo reunir en una sola herramienta funcionalidades requeridas durante el desarrollo.

### 4.1.5 MACROMEDIA, DREAMWEAVER MX

Esta herramienta de edición de páginas Web cuenta con un amplio soporte para distintos formatos, como son los: htm, html, asp, aspx, php, jsp, entre otros. Además de contar con una gran variedad de herramientas que permiten el uso de gráficos, cuenta con las siguientes características:

- Control de cambios
- Apoyo para edición de sitios Web en línea
- Resumen documentado del sitio
- Interfaz segura de comunicación con el Sitio y el editor de hojas Web
- Herramientas de esquematización del sitio
- Historial
- Simplicidad de uso, manejo de tablas, campos Web, Web zone, componentes Web, entre otros

El empleo de este editor de páginas Web se debe al soporte recibido por parte del especialista en diseño gráfico, que colaboró en el desarrollo de la interface gráfica general a la solución Web y a la capacitación recibida para manipular esta interface.

## 4.2 DESARROLLO

La metodología empleada para el desarrollo de la aplicación se encuentra centrada en Módulos, lo que permite realizar el trabajo de desarrollo en varias partes y poder desarrollar en forma paralela la aplicación y tener mayor beneficios para el producto final.

La justificación de esta selección se basa en que permite desarrollar de forma rápida la aplicación, permitiendo que el proyecto se pueda administrar y supervisar con el uso de un control de código que facilita las tareas administrativas. Además que la aplicación basada

en Web permite este tipo de modulación debido a que los componentes de cada módulo se encuentra deslindado en funcionamiento de cada uno de ellos.

Además de tener un ahorro en tiempo de desarrollo se tiene un control en cuanto al desarrollo, es decir, se puede tener un control en los procesos de codificación, pruebas, desarrollo de las interfaces, uso racional de los recursos, puesto a punto de la aplicación , así como facilidad de documentación de la aplicación.

El desarrollo de la solución Web propuesta se lleva a cabo con la generación de módulos que interrelacionados lógicamente y de acuerdo a las necesidades del cliente permite tener el siguiente esquema:

- **Módulo de Validación.** Este módulo presentara una interface para que el usuario final pueda ingresar a la aplicación, permitiendo determinar el tipo de usuario válido así como sus permisos para poder trabajar en los demás módulos.
- **Módulo de Manejo de cuentas.** El manejo de cuentas de usuarios se encuentra administrador en este módulo, permitiendo crear, modificar o delegar permisos.
- **Módulo de Manejo de datos.** El manejo de datos de azolves se encuentra alojado dentro de este módulo, permitiendo el ingreso de datos, actualización o incluso borrado de los datos.
- **Módulo de Consulta.** Dentro de este módulo se realizan la consulta de los datos de azolves, proporcionando resultados de cada uno de los sitios muestreados, con todos los valores registrados de las pruebas de laboratorio así como las características de la toma de la muestra en sitio.
- **Modulo de Análisis de muestras**. Un análisis de los datos correspondientes a cada sitio muestreado, permite tener una evaluación de la calidad de azolves de acuerdo a normas sanitarias, este modulo se encarga de realizar esta validación, pero además con el uso del algoritmo de clasificación de Azolves desarrollado por el Dr. Juan Manuel Méndez Contreras, se puede tener una clasificación del uso de estos azolves.
- **Galería de Imágenes**. Esta parte permite mostrar de forma ordenada un conjunto de imágenes registradas a cada sitio muestreado, permitiendo observar el sitio de análisis con el fin de conocer las dimensiones del sitio muestreado y posibles problemas de la región.
- Módulo de Ayuda y documentación. Parte de la documentación del proyecto de azolves se encuentra alojado en este módulo.

Algunos casos de uso pueden ser los siguientes:

- Ingreso de nuevas cuentas de usuarios que pueden administrar los datos de azolves
- Ingreso de una nueva muestra de azolves, con todas sus características así como resultados de pruebas de laboratorio.
- Consulta de datos de 1 o mas muestras ya registrada, con el fin de cotejar los datos ingresados, con el fin de validar o actualizar los valores registrados
- Consulta del estado final de la valoración de un sitio de estudio, en base a la evaluación de los datos de las muestras previamente ingresados y rectificados.

- Consulta de la información de datos del proyecto de azolves.

Estos son algunos de los casos de uso que se pueden presentar, y es posible la consulta y acceso a alguno(s) de los módulos necesarios para poder realizar tareas de estudio de nuevos o sitios ya registrados.

### 4.2.1 EJECUCIÓN DE HERRAMIENTAS DE DISEÑO Y CONTRUCCIÓN

El desarrollo de la interfaz de las clases se compone de tres elementos, los cuales son:

- Footer
- Header
- Cuerpo de aplicación

Esto nos permite de forma simplificada efectuar cambios de diseño en la interfaz de modo rápido y homogéneo en cada una de las clases que componen la aplicación, para poder realizar esta tarea se recurre al diseño en modo HTML en el cual son declarados los siguientes elementos como parte del cuerpo de la página.

```
<%@ Register TagPrefix="uc1" TagName="Header" Src="./Header.ascx" %>
<%@ Register TagPrefix="uc1" TagName="Footer" Src="./Footer.ascx" %>
<uc1:Header id="Header1" runat="server"></uc1:Header>
<form id="plantilla" method="post" runat="server">
Cuerpo de la aplicación
</form>
<uc1:Footer id="Footer1" runat="server"></uc1:Footer>
```

La construcción en el modo de diseño da como resultado la Figura 4-2 y el resultado final se ve reflejado como la Figura 4-3, esto permite un desarrollo simplificado para los programadores y para el diseñador, además de modular las clases de forma adecuada para posibles modificaciones en el diseño sin alterar el funcionamiento de la aplicación.

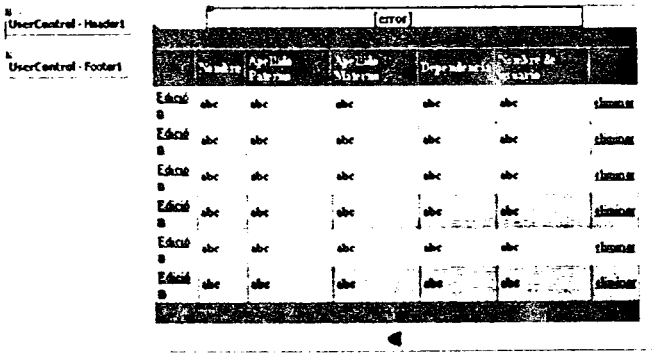



**Figura 4-2 Diseño de clases**

El contenido tanto del "Footer" como del "Header" fueron desarrollados con el empleo de Dreamweaver MX por el experto en diseño gráfico, la Figura 4-2 presenta el diseño final de la interfaz.

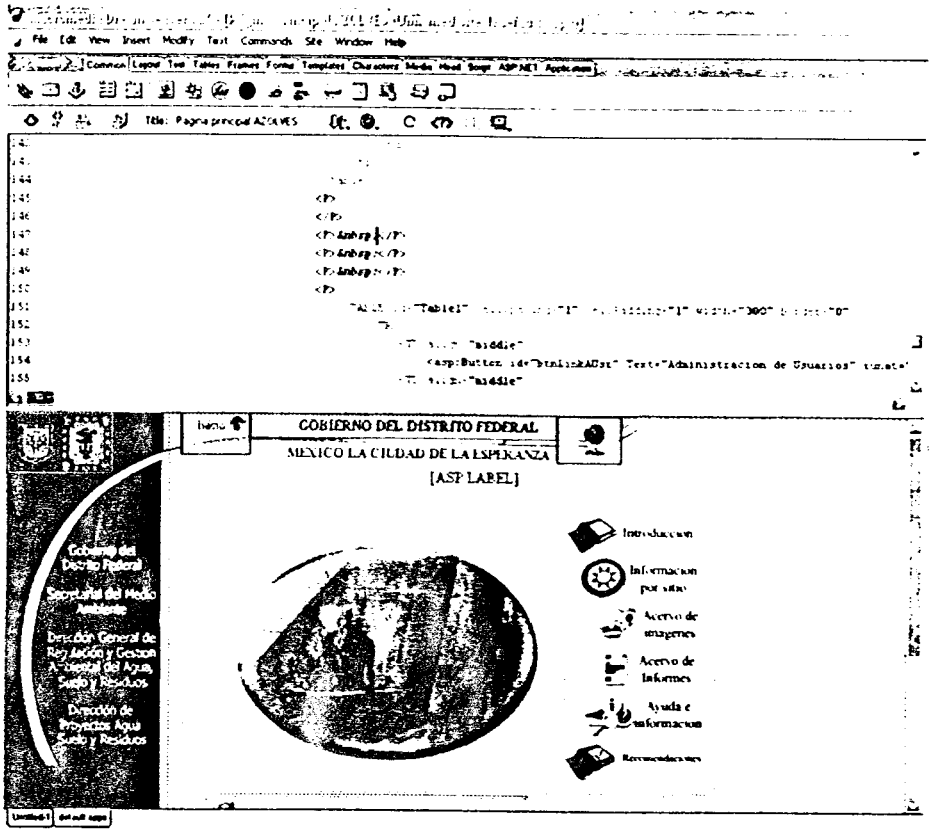



**Figura 4-3 Diseño de Interfaz**

El manejo del "*Footer*" y del "*Header*", conforman el cuadro del diseño final y es, para efectos de implementación, formulado en dos clases separadas e independientes, esto nos permite reducir la cantidad de código a escribir por cada una de las paginas ASPX resultantes, así como tiempo de procesamiento para los equipos clientes.

### 4.2.2 MODELADO DE DATOS

Los datos que se manejan en la aplicación son de distinta naturaleza, originados por el resultado del estudio de los azolves, los cuales son dictados por los procesos de estudio de los investigadores y sus necesidades de análisis. Pero se pueden englobar en los siguientes elementos:

- Carácter descriptivo y cualitativo, que es presentado en texto

- Cantidades físicas que son representadas con valores numéricos, en con el manejo adecuado de unidades, debido a la naturaleza de su origen. (Volumen, masa, peso, etc)
- Imágenes que presentan una fotografía del sitio de estudio

El modelado de los datos de la aplicación se encuentra regido por el orden lógico del tratamiento de los datos, esto originado por las necesidades de estudio de cada muestra y del manejo de datos de cada una de ellas. Como se vio en el Capítulo 2.5 Modelado de la base de datos, los datos se encuentran agrupados en 11 Clases que componen la estructura de datos que alojará los datos registrados por la toma de muestras de azolves y análisis de los mismos.

### 4.2.3 INICIO DE SESIÓN

El módulo de validación se encuentra formado por la clase Logon donde se lleva a cabo el proceso de validación de usuarios, esta clase posee los atributos y métodos mostrados en la Figura 4-4

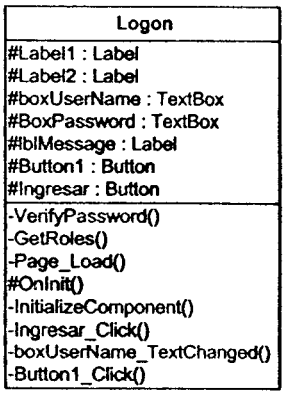



**Figura 4-4 Clase Logon**

El diseño de la interfaz se presenta de la Figura 4-5, donde los atributos de la clase nos permiten tener un acceso simplificado a la aplicación, los *TextBox* correspondientes a *BoxUserName* y *BoxPassword* son elementos de entrada para el usuario, proporcionando el nombre de usuario y la contraseña correspondiente, la clase Logon permite a través de las funciones *VerifyPassword* y *GetRoles*, validar la cuenta del usuario así como verificar el tipo de rol que desempeñará la cuenta proporcionada. Estas funciones y atributos son de carácter protegido debido a la naturaleza de la seguridad de las cuentas y de su tipo de rol así como la información con la que se cuenta.

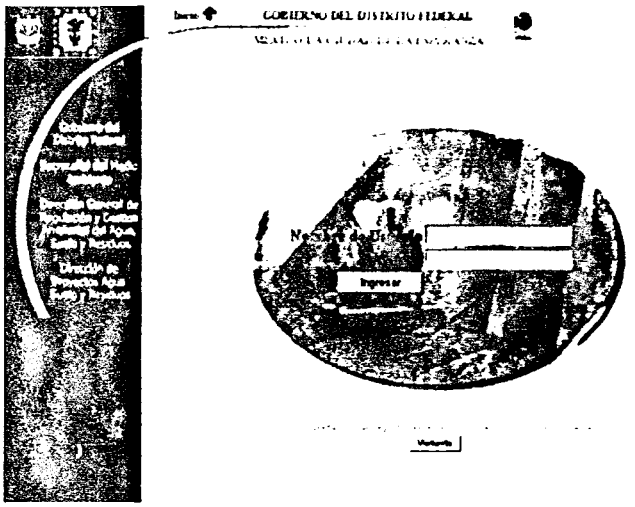


**Figura 4-5 Página de Inicio de Sesión**

Validada la cuenta proporcionada por el usuario se presenta la siguiente clase: *Default*, que permite tener acceso a las distintas operaciones que proporciona la solución Web, dependiendo del tipo de rol a la que pertenece la cuenta proporcionada y validada, se presentan o no las opciones de administración de datos y/o administración de cuentas de usuarios, con base en la Tabla 4-1

**Tabla 4-1 Atributos de cuentas de usuario**

| Usuario | Atributos |
|---|---|
| Administrador | Manejo de Datos, Manejo de Usuarios, consulta de información del proyecto y ayuda |
| Usuario Avanzado | Manejo de Datos, consulta de información del proyecto y ayuda |
| Invitado | Consultar la información del proyecto y ayuda |

La clase *default* de la Figura 4-6 cuenta con una estructura e interface necesarias para poder obtener un menú entre las tareas administrativas para las cuentas de usuarios y otra para la administración de los datos de azolves, por medio de los atributos de liga de botón de *btnLinkAUsr* y de *btnLinkAdat*, correspondientemente.

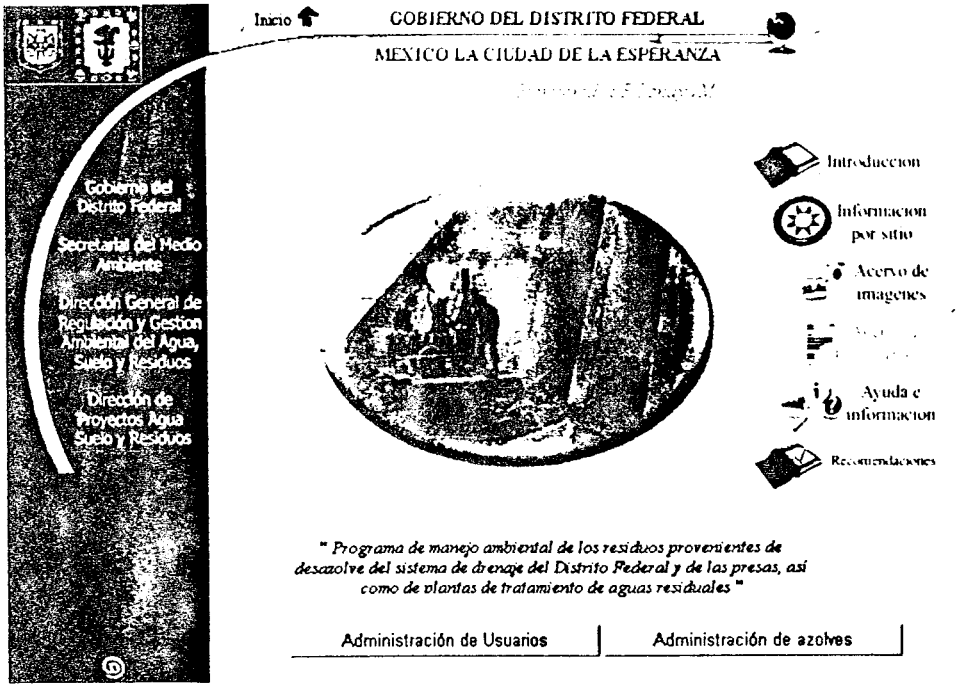



**Figura 4-6 Clase Default**

### 4.2.4 MANEJO DE CUENTAS DE USUARIOS

El registro de usuarios se realiza con el apoyo de la clase Registro, que permite el ingreso de cuentas de usuarios de forma simplificada, debido a la interfaz que sólo limita el ingreso de cierto tipo de datos y requiere, debido a los requisitos del sistema, contar con los elementos. La interfaz para el acceso al manejo de cuenta de usuarios se muestra en la Figura 4-7, la cual contiene el acceso a las clases:

- Registro (Registro de Usuarios)
- DatosAtr (Manejo de cuentas de Usuarios)
- DatosUsr (Edición de permisos a cuentas de Usuarios)
- UpdateLlave (Cambio de contraseña)
- Retorno a la clase del Menú principal

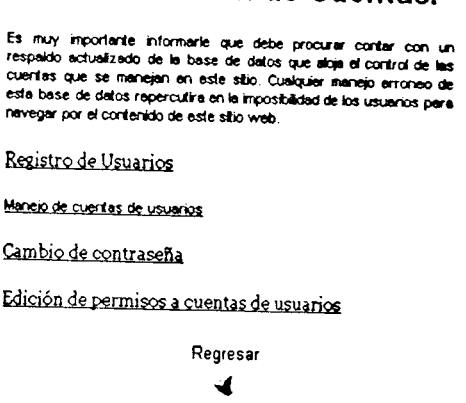



**Figura 4-7 Interfaz de Manejo de usuarios**

### 4.2.4.1 CLASE REGISTRO

Esta clase mostrada en la Figura 4-8 contiene los atributos *TextBox* de tipo *WebControls* y de visibilidad protegida, que permiten el medio de entrada por parte del usuario para poder ingresar la información solicitada para el registro de usuarios.

El ingreso de los datos proporcionados en los cuadros de texto (*TextBox*), se encuentra a cargo de la operación *Botton1_Click*, la cual lleva a cabo la validación de la contraseña proporcionada por medio de una comparación de contenido de #password de #Password2 y *txtPassword*, siendo que para el caso "falso" manda una advertencia de corrección de datos en *password*.

Para el caso "verdadero" se realiza el acceso a la base de datos por medio de una conexión ADO, lo que permite establecer un canal de comunicación entre la aplicación y la base de datos, realizado este acceso se lleva a cabo el envío de datos hacia la base de datos con el control del Cliente ASP, que realiza el depósito de datos en las tablas correspondientes de la base de datos.

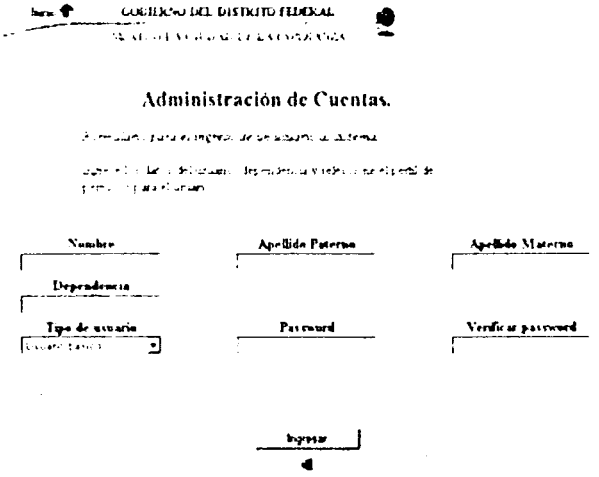



**Figura 4-8 Clase Registro e Interfaz**

La interfaz de la clase Registro contiene los 6 *TextBox* para el ingreso de la información del usuario, 1 *Botton* para el evento de acceso de datos, y un *DropDownList* que contiene el listado de los tipos de usuarios que se pueden seleccionar.

Esta interfaz permite ingresar los datos de usuario solicitados, proporcionar una selección del tipo de usuario, así como el registro a la base de datos de Usuarios. Este medio simplifica de modo visual el ingreso de nuevos usuarios, garantizando la creación de la cuenta de Usuario y uso inmediato de la misma.

### 4.2.4.2 CLASE DATOSATR

Esta clase contiene la estructura mostrada en la Figura 4-9 y permite la edición de los atributos de las cuentas registradas, permitiendo al administrador delegar atributos de usuario avanzado o usuario básico, con el fin de simplificar tareas de manejo de cuentas y manejo de datos.

El atributo *WebControls.TextBox boxUserName* y el atributo *WebControls.Button* buscar y la función Buscar_Click() permiten ingresar el nombre de Usuario y búsqueda de la cuenta proporcionada con el fin de poner en deposición inmediata los atributos de dicha cuenta y poder editar el tipo de permisos para ella.

Los atributos de *WebControls.Label* Blanco nos indicarán los atributos de la cuenta solicitada, permitiendo al Administrador de cuentas, saber si es o no, la correspondiente a la persona a la cual se desea cambiar el privilegio para el acceso a la solución Web.

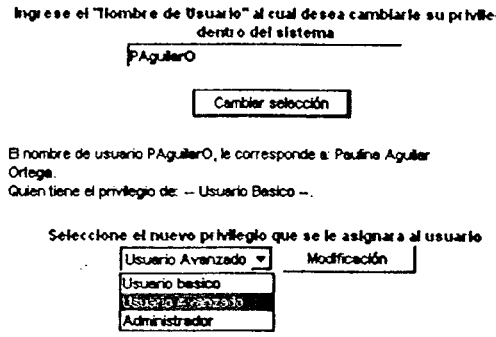



**Figura 4-9 Clase DatosAtr e Interfaz**

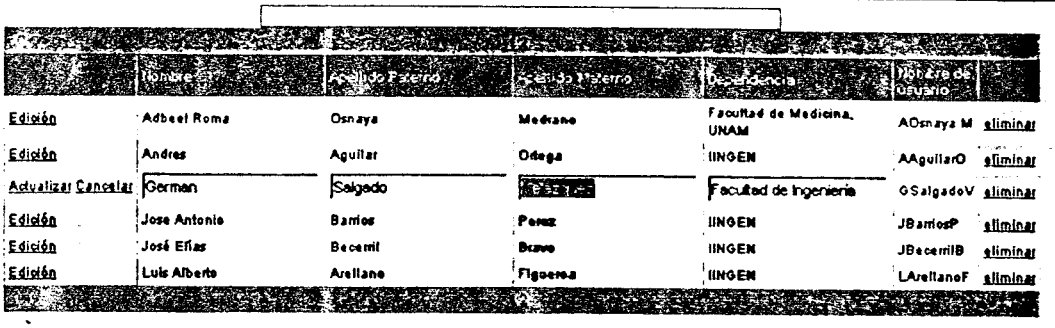
En caso de requerir un cambio de selección, se proporciona una nueva búsqueda con el atributo de *WebControls.Button* Otro y la función Otro_Click(), el cual permite realizar una nueva búsqueda en la base de datos de usuarios.

Para la modificación de los privilegios de la cuenta seleccionada se emplea un *WebControls.DropDownList* para mostrar las posibles opciones de privilegios. Y para poder realizar este cambio se emplea el *WebControls.Button* Cambiar con el empleo de la función Cambiar_Click() que toma los atributos de la cuenta seleccionada, la selección de tipo de privilegio de la lista, y realiza la actualización a la base de datos de usuarios, por medio de *AttribUpdate.*

### 4.2.4.3 CLASE DATOSUSR

Esta clase pose una interfaz que simplifica el control de las cuentas de los usuarios y permite dentro de sus tareas fundamentales la edición, eliminación de una forma simplificada debido a las funciones que son aplicadas al *WebControls.DataGrid DataGrid1*, que de forma interna simplifican el acceso a la base de datos y de forma segura se llevan a cabo los cambios seleccionados por medio de la las funciones: *DataGrid1_EditCommand, DataGrid1_CancelCommand, DataGrid1_UpdateCommand, DataGrid1_DeleteCommand*

La estructura de datos del *DataGrid* permite la simplificación de las operaciones así como la presentación de datos, lo que nos permite actualizar los datos de usuarios de forma rápida, además de simplificar futuros cambios en cuanto al número de atributos y tipo, ya que con sólo pocos cambios en el número de registros de petición del *DataGrid* es posible realizar un manejo de más atributos.



| | Nombre | Apellido Paterno | Apellido Materno | Dependencia | Nombre de usuario | |
|---|---|---|---|---|---|---|
| Edición | Adbeel Roma | Osnaya | Medrano | Facultad de Medicina, UNAM | AOsnaya M | eliminar |
| Edición | Andres | Aguilar | Ortega | IINGEN | AAguilarO | eliminar |
| Actualizar Cancelar | German | Salgado | [illegible] | Facultad de Ingenieria | GSalgadoV | eliminar |
| Edición | Jose Antonio | Barrios | Perez | IINGEN | JBarriosP | eliminar |
| Edición | José Elías | Becerril | Bravo | IINGEN | JBecerrilB | eliminar |
| Edición | Luis Alberto | Arellano | Figueroa | IINGEN | LArellanoF | eliminar |

**Figura 4-10 Clase Datosusr e Interfaz**

El resultado final se puede ver en la Figura 4-10, donde el acceso final al *DataGrid1* permite de modo gráfico tener un manejo de las cuentas de usuarios, teniendo las funciones edición, borrado, actualización o cancelar la edición.

### 4.2.4.4 CLASE UPDATELLAVE

Para las tareas administrativas del manejo de cuentas de usuarios se requiere de un módulo de cambio de contraseña como se muestra en la Figura 4-11, ya que el uso de datos y manejo de cuentas requieren de un acceso seguro, es por eso que sólo el administrador puede realizar este tipo de cambios, y para ello se diseña la siguiente clase *UpdateLlave*.

Las funciones *VerifyPassword()* y *GetRoles()*, son empleadas para validar al administrador que realizará un cambio de contraseña sobre otra cuenta de usuario, por medio de esta validación se verifica la existencia de la cuenta y de los atributos con los que se cuenta para poder realizar este cambio en otras cuentas de usuarios.

La función Buscar_Click() y el *WebControls.TextBox* Usuario permiten ingresar la cuenta del usuario deseado y busca dentro de la base de datos de usuarios la cuenta solicitada y realiza una lectura de los atributos, depositando estos datos en el *WebControls.Label* Resultado para poder confirmar el resultado de la búsqueda y poder realizar cambios en la contraseña. Esta acción será controlada por la función Continuar_Click() del *WebControls.Button* Continuar, que solicitará al Administrador si el proceso continúa o no. Siendo para el caso de "no", el poder llevar a cabo una nueva búsqueda o regresar al menú principal de manejo de cuentas, y para el caso de "continuar" se lleva a cabo el ingreso de la nueva contraseña en los *WebControls.TextBox NewPassword* y *RNewPassword*, que son los medios de entrada para los parámetros de la función Cambio_click().

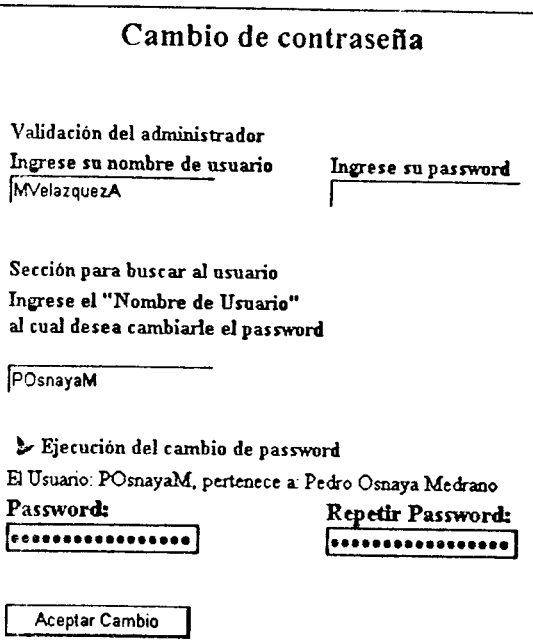



**Figura 4-11 Clase UpdateLlave e Interfaz**

La función Cambio_click(), se encarga de realizar los cambios de la cuenta seleccionada directamente en la base de almacenamiento, con el apoyo de los atributos y comandos de SQL que se emplean para obtener la conexión, consulta y actualización de los datos requeridos, siendo para este caso el cambio de contraseña.

### 4.2.5 MANEJO DE DATOS

El manejo de los datos del Proyecto de Azolves, requiere de un control y cuidado en cuanto a la seguridad de datos y al tipo de información que se almacena, es por eso que las siguientes clases también requieren de un método se seguridad, la clase OpcionesDatos de la Figura 4-12 describe las siguientes características:

- Manejo de Clases de Muestreo.
- Manejo de Sitios de Muestreo.
- Manejo de los datos de las Muestras de Azolves analizadas.
- Manejo de la caracterización de las Muestras.

Estas características son presentadas de una forma simple por medio de ligas de acceso a los módulos que se encargan de presentar las herramientas necesarias para el ingreso de datos de las clases, sitios y muestras a la base de datos, para su posterior análisis y estudio.

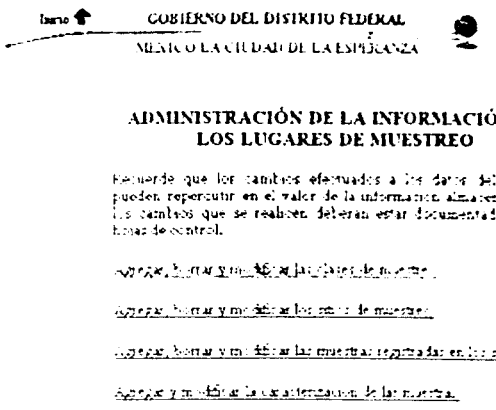



**Figura 4-12 Clase OpcionesDatos**

La validación y acceso a esta clase, se lleva a cabo por medio de la función *Page_Load()* que checa el tipo de rol que posee la cuenta de usuario proporcionada al sistema, esto permitirá o no el despliegue las opciones para el manejo de datos, además este método se presentará en las siguientes clases, con el fin de tener una seguridad en cuanto al identificador de quién está laborando con los datos.

### 4.2.5.1 CLASE ADMCLASES

El manejo de Clases de Muestreo se encuentra formulado en la clase *AdmClases*, donde se cuenta con la estructura de la Figura 4-13, la cual realiza una conexión a la base de datos, correspondiente a la tabla Clases, donde los atributos de "*clasename*" y "*clasedescription*" son los parámetros de interés, ya que el manejo de azolves requiere de una clasificación de acuerdo al tipo de origen de muestreo, como puede ser presa, lago, río, entre otros.

La interfaz mostrada permite por medio del *WebControls.DataGrid DataGrid1* interactuar de modo gráfico con el usuario final de forma simplificada en cuanto a tareas de edición del tipo de Clases de muestreo, ingreso de clases o borrado de estas mismas por medio de las funciones:

- *DataGrid1_UpdateCommand()*
- *DataGrid1_DeleteCommand()*
- *DataGrid1_EditCommand()*
- *DataGrid1_CancelCommand()*

Y para el caso de crear una nueva clase se hace empleo del un formulario compuesto por:

- *WebControls.TextBox NewClase*
- *WebControls.TextBox NewDescrip*
- *WebControls.Button Button2*
- *WebControls.Button Button3*

Que permiten la captura de los datos de la nueva clase, nombre y descripción, así como su ingreso en la base con el empleo de *Button2* y la función *Button2_Click()*, que realiza el ingreso del registro en la base de datos, permitiendo para el manejo de sitios y muestras una nueva clasificación de Tipo de Sitio disponible.

| Edirión | Clase | Descripcion de la clase | Eliminar clase |
|---|---|---|---|
| Edición | Planta de Tratamiento | Una planta de tratamiento de Aguas residuales | Eliminar |
| Edición | Presa | Una presa es un lugar... | Eliminar |
| Edición | Lagunas | Una laguna es... | Eliminar |
| Edición | Cauces y rios | Categoria que abarca causes y rios | Eliminar |
| Edición | Estaciones de Transferencia | Una estacion de transferencia de azolves y lodos | Eliminar |
| Edición | Vasos deRegula | Un vaso de regulacion es... | Eliminar |

Agregar nueva clase

**Figura 4-13 Interfaz de la Clase AdmClases**

### 4.2.5.2 CLASE ADMLUGARES

El manejo de los lugares o sitios de muestreo, contienen una caracterización simple, donde los siguientes parámetros son registrados en un formulario que se muestra en la Figura 4-14

- Clave de lugar
- Clase de lugar
- Nombre del lugar
- Dirección
- Mapa

El cargado de esta clase presenta por medio de un *DataGrid*, un resumen de los lugares ya registrados, permitiendo su manejo, ya sea su Clave, nombre, dirección o mapa, permitiendo la corrección de los datos ya ingresados.

Por medio de un *ControlBoton* se presenta el formulario para el ingreso del nuevo lugar el cual permitirá al usuario el llenado de los parámetros solicitados.

El control para la presentación de la interfaz entre el *DataGrid* y el Formulario, se lleva a cabo con el *ControlBoton*, el cual habilita la visibilidad del formulario y deshabilita el del *DataGrid*, esto permite el empleo de la misma página Web

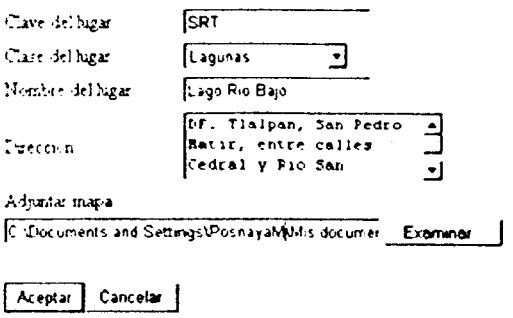



**Figura 4-14 Clase AdmLugares, Interfaz de sitios de muestreo**

Para el manejo del atributo de Clase de Sitio de muestra se emplea un *RadioButtonList*, que presenta el listado de las clases que se encuentran alojadas, y la selección de alguna de esta muestra los lugares que contiene esta clasificación, permitiendo la edición de este sitio por medio de la función *Button_Click* que realizará la actualización en la base de datos del atributo Clase de Lugar del Lugar de muestreo que se encuentra seleccionado.

### 4.2.5.3 CLASE ADMMUESTRAS

La administración de las muestras de azolves se lleva a cabo con el empleo de un formulario que permitirá el manejo de los elementos que componen la caracterización de una muestra, siendo estos:

- Clase de sitio de muestreo
- Sitio de muestreo
- Datos del sitio de toma de la muestra
- Ingreso de una nueva muestra

Para el manejo de las Clases de Sitio del proyecto, se emplea un conjunto de *WebControls.RadioButtonList*, que de forma automática presentan la lista de Clases de sitios que existen en la base de datos, permitiendo realizar una selección adecuada para el manejo de las muestras.

De la misma forma se presentan los sitios de muestreo correspondientes a la Clase de Sitios, estos Sitios son los correspondientes a la Clase de Sitio de muestreo, por lo que el manejo de las muestras de azolves son almacenadas de forma correcta en su respectiva clasificación.

Esta interfaz presenta el manejo de un *WebControls.Calendar* para el ingreso del atributo de fecha, facilitando al usuario final mediante una interfaz más agradable y funcional, como un punto de referencia para el manejo de fechas del registro y toma de las muestras.

Para el manejo de datos delimitados se presenta el uso de *WebControls.DropDownList* que permite el ingreso de datos que son delimitados por la naturaleza del tipo de dato y para evitar error de lectura e ingreso de datos no deseados.

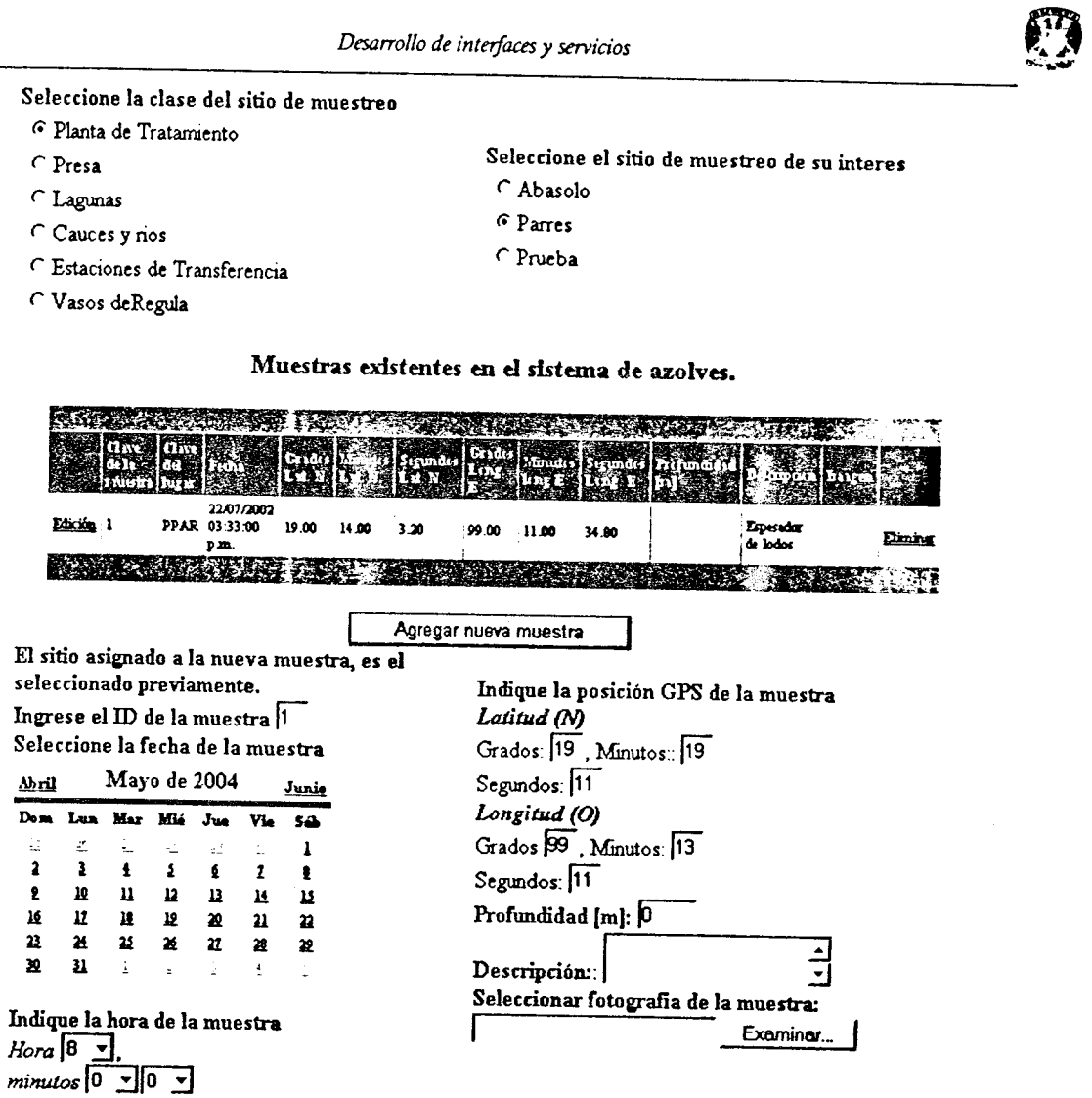



**Figura 4-15 Interfaz de la clase AdmMuestras**

Se presentan por medio de un *WebControls.DataGrid* las Muestras que corresponden al Sitio de Muestreo seleccionado, permitiendo realizar un manejo de los datos, o realizar el ingreso de un nuevo registro por medio del formulario

### 4.2.6 CLASE SITIOS

La clase sitios permite la selección gráfica de cada uno de los sitios de estudio, permitiendo obtener información correspondiente a cada uno de los sitios así como los datos correspondientes a cada una de las muestras registradas en los sitios de muestreo, como se ve en la Figura 4-1, donde los sitios poseen una nomenclatura para poder ser localizados con facilidad.

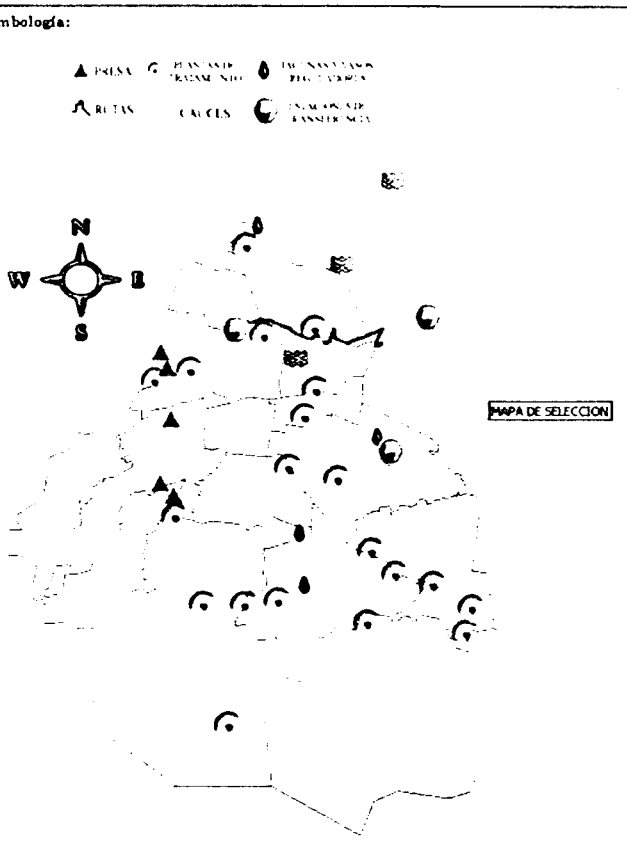
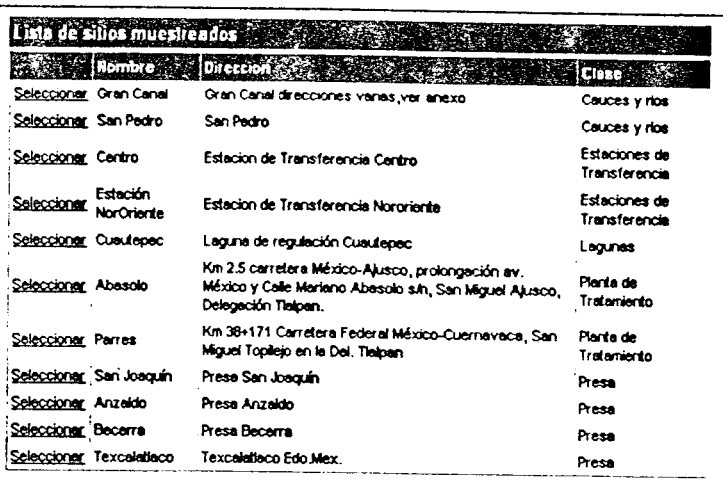



**Figura 4-1 Interfaz de Sitios**

Esta clase cuenta con elementos de selección gráfica y ligas, que hacen la solicitud a las clases de Sitiosmuestreados.aspx o seleccion.aspx de los sitios factibles de ser analizados. Las siguientes clases al igual que esta poseen un sistema de validación para poder ser consultadas, como fue para el caso de las clases de manejo de usuarios y de datos de azolves, esta parte de información del estudio de azolves también requiere de la validación del tipo de usuario y del tipo de rol que desempeñará en la aplicación, por lo que se hace uso de la función ***private void Page_Load(object sender, System.EventArgs e)***, que al cargado de esta página permite revisar estos atributos.

### 4.2.6.1 SITIOS MUESTREADOS

Para la presentación de información de los sitios, se presenta la siguiente clase que permite, por medio de *DataGrid*, seleccionar alguno de los sitios de interés, este *DataGrid* realiza una consulta a la base de datos, recuperando los sitios de estudio así como sus atributos de nombre, identificador y tipo de sitio, esto permite mostrar al usuario los datos actualizados.



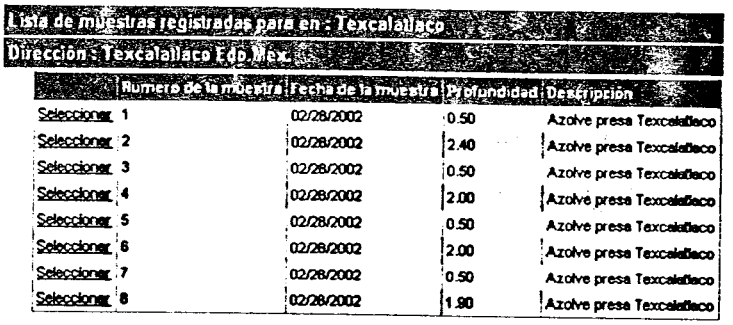
Lista de sitios muestreados

| | Nombre | Direccion | Clase |
|---|---|---|---|
| Seleccionar | Gran Canal | Gran Canal direcciones varias,ver anexo | Cauces y rios |
| Seleccionar | San Pedro | San Pedro | Cauces y rios |
| Seleccionar | Centro | Estacion de Transferencia Centro | Estaciones de Transferencia |
| Seleccionar | Estación NorOriente | Estacion de Transferencia Nororiente | Estaciones de Transferencia |
| Seleccionar | Cuautepec | Laguna de regulación Cuautepec | Lagunas |
| Seleccionar | Abasolo | Km 2.5 carretera México-Ajusco, prolongación av. México y Calle Mariano Abasolo s/n, San Miguel Ajusco, Delegación Tlalpan. | Planta de Tratamiento |
| Seleccionar | Parres | Km 38+171 Carretera Federal México-Cuernavaca, San Miguel Topilejo en la Del. Tlalpan | Planta de Tratamiento |
| Seleccionar | San Joaquín | Presa San Joaquín | Presa |
| Seleccionar | Anzaldo | Presa Anzaldo | Presa |
| Seleccionar | Becerra | Presa Becerra | Presa |
| Seleccionar | Texcalatlaco | Texcalatlaco Edo.Mex. | Presa |

**Figura 4-16 Interfaz de Sitios Muestreados**

### 4.2.6.2 INTROSITIOS

La clase Intrositios está formada fundamentalmente por un *DataGrid* que consulta la base de datos y presenta las muestras registradas para el sitio seleccionado en la clase SitiosMuestreados, presentando algunas características de cada muestra, como son su identificador ID, Fecha de muestreo, profundidad de la muestra así como su descripción. El acceso y consulta hacia la base se realiza por medio de un *SqlDataAdapter* y un *SqlConnection*, esto permite depositar en el *DataGrid* los datos seleccionados y establece la interfaz con el fin de mostrar las muestras correspondientes al sitio seleccionado.



Lista de muestras registradas para en : Texcalatlaco

Dirección : Texcalatlaco Edo.Mex.

| | Numero de la muestra | Fecha de la muestra | Profundidad | Descripcion |
|---|---|---|---|---|
| Seleccionar | 1 | 02/28/2002 | 0.50 | Azolve presa Texcalatlaco |
| Seleccionar | 2 | 02/28/2002 | 2.40 | Azolve presa Texcalatlaco |
| Seleccionar | 3 | 02/28/2002 | 0.50 | Azolve presa Texcalatlaco |
| Seleccionar | 4 | 02/28/2002 | 2.00 | Azolve presa Texcalatlaco |
| Seleccionar | 5 | 02/28/2002 | 0.50 | Azolve presa Texcalatlaco |
| Seleccionar | 6 | 02/28/2002 | 2.00 | Azolve presa Texcalatlaco |
| Seleccionar | 7 | 02/28/2002 | 0.50 | Azolve presa Texcalatlaco |
| Seleccionar | 8 | 02/28/2002 | 1.90 | Azolve presa Texcalatlaco |

**Figura 4-17 Interfaz de Intrositios**

La selección de las muestras proporciona su identificador como parámetro para la función de carga de datos para la clase ListaMuestras, que adquiere los atributos de ID_Lugar y ID_Muestra para la consulta requerida.

### 4.2.6.3 LISTAMUESTRAS

Esta clase presenta en resumen algunas características de la tabla Muestras, como son posición geográfica, profundidad o notas realizadas por el personal de muestreo, con uso de de *SqlConector* y de un *SqlAdapter* se permite la consulta de cada uno de estos elementos y

son colocados en la interfaz final por medio de etiquetas que presentan los valores en la página aspx.

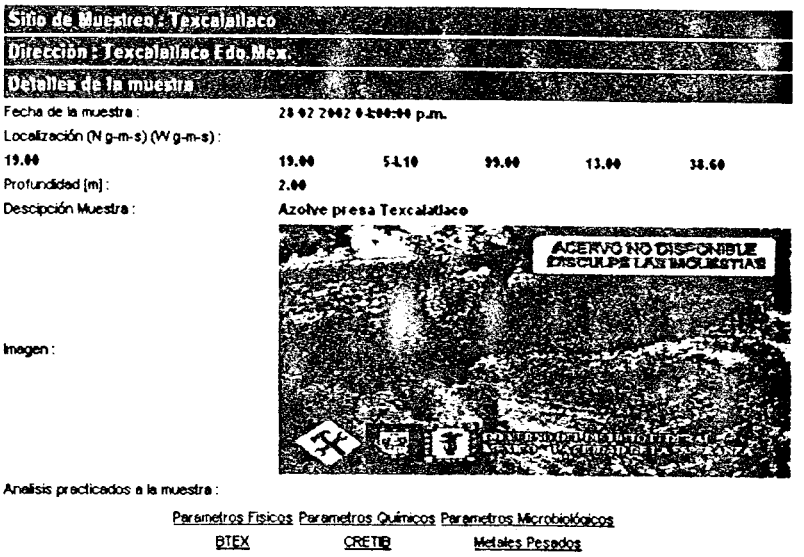



**Figura 4-18 Interfaz de Listamuestras**

Esta clase permite la presentación de una imagen del sitio de muestreo, la cual es tomada por medio de una referencia directa del archivo de imagen, por medio de un recuadro de imagen que dentro de su variable "*image*" hace referencia a la localización de esta por con el empleo de una ruta generada por concatenación del nombre del archivo que se encuentra alojado en el base de datos y la ruta obtenida que hace referencia al fólder contenedor de las imágenes.

### 4.2.6.4 DETALLESMUESTRA

Los detalles de las muestras se presentan como un resumen de las características de las muestras, para cada una de las pruebas de laboratorio. Estas consultas son proporcionadas con el apoyo de *SqlClient.SqlDataAdapter*, que selecciona de la base de datos los registros de la tabla correspondiente de cada parámetro o prueba deseada.

Para los detalles de las muestras se realiza la consulta para:

- Parámetros Físicos
- Parámetros Químicos
- Parámetros Microbiológicos
- BTEX
- CRETIB
- Metales Pesados

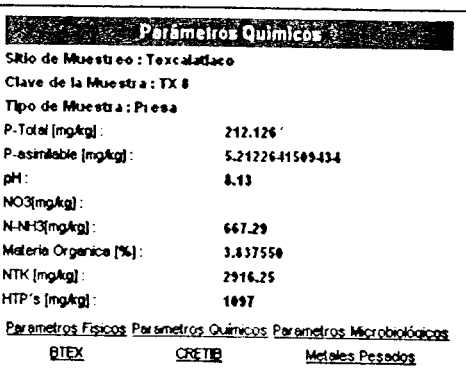



**Figura 4-19 Interfaz de Detallesmuestra**

Cada una de estas consultas son realizadas de forma independiente, con el empleo de *SqlClient.SqlDataAdapter* se realiza la comunicación necesaria entre la base y la interfaz final permitiendo mediante etiquetas mostrar los valores de cada uno de los registros de cada muestra, como se muestra en la Figura 4-19.

El detalle de la carga de datos se realiza con la función *LoadData()*, la cual esta definida de la siguiente forma:

**private void LoadData()**

```
        {
sqlSelectCommand1.Parameters[0].Value = System.Int32.Parse(IDMuestra.ToString());
        sqlSelectCommand1.Parameters[1].Value = IDLugar.ToString();
        MB.Fill(dsPMicroB1);
        lblSitio.Text = dsPMicroB1.Tables[0].Rows[0]["Nombre"].ToString();
        lblMuestra.Text = IDLugar.ToString() + IDMuestra.ToString();
        lblClase.Text = dsPMicroB1.Tables[0].Rows[0]["Clase"].ToString();
        lblColiformes.Text = dsPMicroB1.Tables[0].Rows[0]["Coli_Fecales"].ToString();
        lblSalmonella.Text = dsPMicroB1.Tables[0].Rows[0]["Salmonella"].ToString();
lblHelminto.Text = dsPMicroB1.Tables[0].Rows[0]["Huevos_Helminto"].ToString();
        }
```

Donde el adaptador "MicroB1 para este caso, recopila los valores para cada valor de la tabla seleccionada y deposita estos valores en etiquetas para proporcionar la interfaz final en las páginas apsx.

### 4.2.7 ANÁLISIS DE MUESTRAS

La clase Análisis de Muestras, realiza una evaluación basada en el algoritmo descrito en el capitulo 2, donde se lleva a cabo la clasificación del sitio de muestreo de acuerdo al cumplimiento de las normas de calidad de azolves. Este análisis recopila la información de los registros de las muestras de todas las tablas que componen los datos de la muestra y son comparados de acuerdo al algoritmo con el criterio de valores definidos por las normas y al criterio establecido por el Grupo de Tratamiento y Reuso del Instituto de Ingeniería. Los sitios factibles a ser analizados son listados en la clase seleccion.aspx que emplea un

*DataGrid* para su presentación, del mismo modo es empleado un *SqlConection* para poder realizar la conexión y un *SqlAdapter* para integrar los registros del *DataGrid* y ser presentados en la interfaz.

Muestras Factibles de analizar

| | Nombre | Direction | Clase |
|---|---|---|---|
| Ejecutar análisis | Anzaldo | Presa Anzaldo | Presa |
| Ejecutar análisis | Becerra | Presa Becerra | Presa |
| Ejecutar análisis | Cuautepec | Laguna de regulación Cuautepec | Lagunas Vasos Regula |
| Ejecutar análisis | Centro | Estacion de Transferencia Centro | Estaciones Transf |
| Ejecutar análisis | Nororiente | Estacion de Transferencia Nororiente | Estaciones Transf |
| Ejecutar análisis | Oriente | Estacion de Transferencia Oriente | Estaciones Transf |
| Ejecutar análisis | Gran Canal | Gran Canal direcciones varias,ver anexo | Cauces y rios |
| Ejecutar análisis | Laguna Mayor | Laguna de regulación Mayor | Lagunas Vasos Regula |
| Ejecutar análisis | Cienaga Chica | Laguna de regulación Ciénega Chica | Lagunas Vasos Regula |
| Ejecutar análisis | Remedios | Río de los Remedios direcciones varias,ver anexo | Cauces y rios |
| Ejecutar análisis | San Joaquín | Presa San Joaquín | Presa |
| Ejecutar análisis | San Lucas | Vaso de regulación San Lucas | Lagunas Vasos Regula |
| Ejecutar análisis | Tecamachalco | Presa Tecamachalco | Presa |
| Ejecutar análisis | Texcalatlaco | Texcalatlaco Edo.Mex. | Presa |

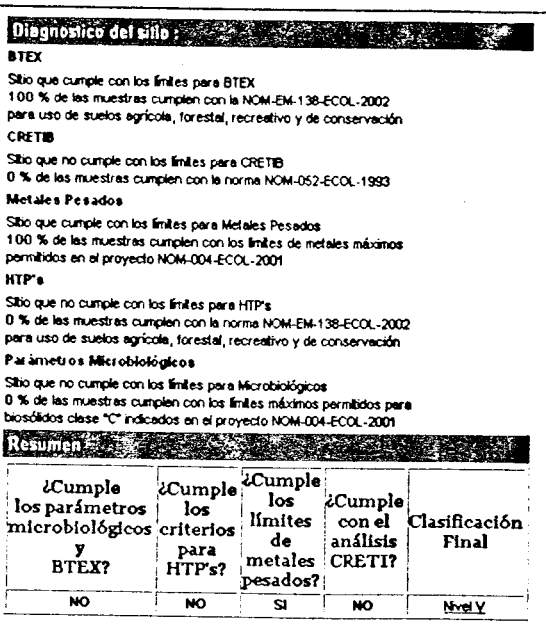
**Figura 4-20 Análisis de muestras**

La ejecución del análisis de los sitios se realiza en el servidor lo que implica que el cliente obtiene exclusivamente los resultados finales es decir, sólo obtiene los resultados y no tiene acceso a los datos de azolves o a los parámetros de criterio para la clasificación de la muestra en cuestión.

Cada resultado es evaluado por una clase de tipo publico y estático, lo que permite evaluar por separado las pruebas de BTEX, CRETIB, Metales Pesados, HTP's y los parámetros microbiológicos, esto permite simplificar la generación de esta interfaz donde para efectos de mantenimiento y de actualización se pueden realizar cambios de los criterios de acuerdo a las normas de calidad de azolves.

El diagnóstico final se presenta en la interfaz mostrada en la Figura 4-21, con el apoyo de la función Evaluar() se realiza una conexión a la base de datos y con el empleo de 5 *SqlAdapter* se realizan las consultas recesarias para extraer los datos para la evaluación de acuerdo al algoritmo de calidad de azolves.

Se presenta para cada una de las pruebas un resultado porcentual así como su aprobación en cuanto al cumplimiento a la norma a la que implica, además de presentar su clasificación final.



**Diagnóstico del sitio :**

**BTEX**

Sitio que cumple con los límites para BTEX
100 % de las muestras cumplen con la NOM-EM-138-ECOL-2002 para uso de suelos agrícola, forestal, recreativo y de conservación

**CRETIB**

Sitio que no cumple con los límites para CRETIB
0 % de las muestras cumplen con la norma NOM-052-ECOL-1993

**Metales Pesados**

Sitio que cumple con los límites para Metales Pesados
100 % de las muestras cumplen con los límites de metales máximos permitidos en el proyecto NOM-004-ECOL-2001

**HTP's**

Sitio que no cumple con los límites para HTP's
0 % de las muestras cumplen con la norma NOM-EM-138-ECOL-2002 para uso de suelos agrícola, forestal, recreativo y de conservación

**Parámetros Microbiológicos**

Sitio que no cumple con los límites para Microbiológicos
0 % de las muestras cumplen con los límites máximos permitidos para biosólidos clase "C" indicados en el proyecto NOM-004-ECOL-2001

**Resumen**

| ¿Cumple los parámetros microbiológicos y BTEX? | ¿Cumple los criterios para HTP's? | ¿Cumple los límites de metales pesados? | ¿Cumple con el análisis CRETI? | Clasificación Final |
|---|---|---|---|---|
| NO | NO | SI | NO | Nivel V |

Figura 4-21 Diagnóstico

## 4.2.8 GALERIAS

Este módulo presenta una la recopilación de fotografías que fueron obtenidas en el momento de que las muestras de azolves fueron tomadas. Se presenta en la figura siguiente la agrupación de los tipos de sitios disponibles, los cuales presentan una agrupación de los sitios que fueron muestreados.

Se establece para cada tipo de sitio de muestreo una lista de las fotografías disponibles, que se encuentran registradas en una tabla de contenido auxiliar, que aloja el identificador del sitio muestreado, como identificador, descripción de la imagen y nombre del archivo correspondiente.

Para poder realizar el despliegue de la imagen de forma correcta, se hace uso de un cuadro de imagen que permite mostrar cada grupo de fotografías correspondientes a cada sitio. Para poder direccional el archivo indicado de cada fotografía se hace uso de una concatenación del nombre del archivo correspondiente con la ruta lógica de la carpeta contenedora. Esto permite de forma automática obtener una referencia de cada archivo y de su contenido y descripción.

Las consideraciones que se tomaron para esta galería son las siguientes:

- Resolución de 92 pix/pulgada
- Dimensión de 450x300 (Formato horizontal)
- Dimensión de 450x600 (Formato vertical)
- Tamaño de archivo de 35Kb a 50Kb

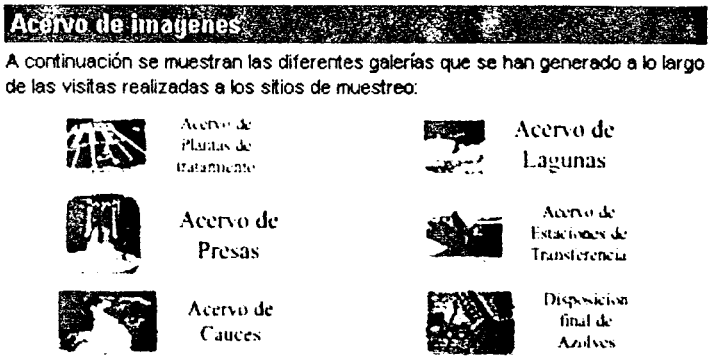



**Figura 4-22 Menú de galería**

Cada una de las galerías dispuestas permite listar los sitios que integran esta clasificación y presenta una interfaz como se muestra en la

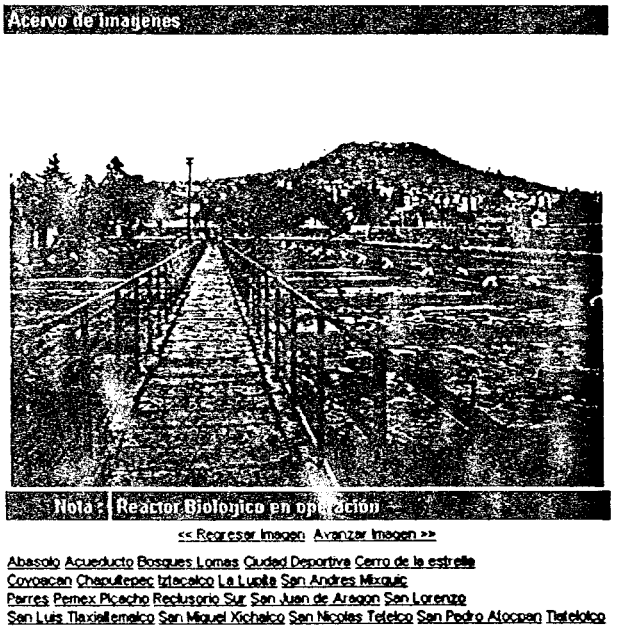



**Figura 4-23 Colección de fotografías**

### 4.2.9 AYUDA E INFORMES

Los informes y texto de apoyo para la consulta de información del sitio, contienen una interfaz simple. Dicha información es alojada en páginas ASPX, con el fin de integrar esquemas de seguridad en cuanto a su consulta y a su modificación. El diseño para la presentación de esta información concuerda con el esquema propuesto en las demás páginas y clase de trabajo en todo el sitio, lo que presenta una homogenización y la posibilidad de cambiar de diseño de forma rápida y segura.

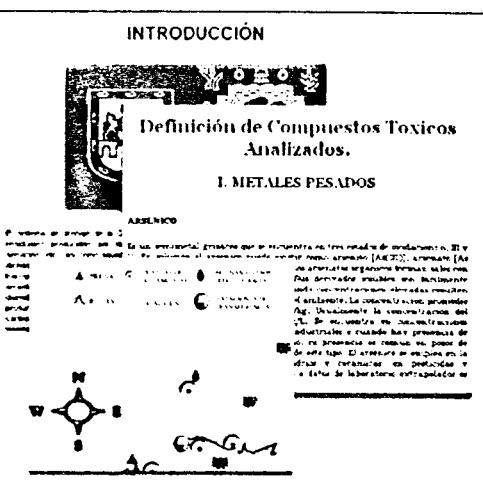



**Figura 4-24 Ayuda e Informes**

El contenido de la información, manejo, uso y actualización es responsabilidad, de acuerdo al convenio entre el Gobierno de la Ciudad de México y la Subdirección de Ingeniería Ambiental del Instituto de Ingeniería, de las partes correspondientes de estudio, adquisición y manejo, es por ello que se presenta de esta forma la información, para que una persona capacitada previamente pueda realizar esta tarea con o sin el apoyo del administrador del sistema.

La presentación de la ayuda para el uso de la aplicación Web, se presenta también en formato de texto en paginas ASPX, con el fin de integrar los mismos criterios de seguridad y poder delegar permisos a las personas que trabajarán en la aplicación o sólo realizarán consultas de información o tareas de administración de datos.

# CAPÍTULO 5 FUTURO DE LA APLICACIÓN

La solución Web propuesta y desarrollada para este proyecto cuenta con una serie de tecnologías nuevas o recientes con el fin de que dicha solución no solo sea capaz de resolver un problema sino que además sea una solución durable y con capacidad de escalarse en un futuro. Entre las tecnologías más importantes se encuentran ".NET", "XML" y el empleo de los "SERVICIOS WEB".

Debido a que este es un proyecto desarrollado a partir de requerimientos previamente establecidos, la funcionalidad que presenta el producto final refleja el cumplimiento de las expectativas del cliente, con lo cual queda demostrado que ésta fue la mejor solución y que podrá servir de base para el desarrollo de otras soluciones a problemas similares, incluso en otras ciudades con la misma problemática ambiental Durante el desarrollo de la solución, se exploró la integración de funcionalidades no mencionadas en los requerimientos técnicos, la integración a futuro de tecnologías y funcionalidades ofrecerán un panorama más amplio de servicios, los cuales incrementarán el beneficio que pueda ofrecer esta solución. El presente capitulo sentará una propuesta del rumbo que puede tomar esta aplicación con el paso del tiempo.

## 5.1 FUTURO DE LAS APLICACIONES N-CAPAS

El diseño de la presente solución Web se llevó a cabo de manera planeada con el fin de obtener una aplicación exitosa y con miras a poder expandirse y mejorar su funcionalidad en una etapa posterior es por ello que se escogió la arquitectura de n-capas y de esta forma al dividir la aplicación en partes lógicas fue mucho más fácil el generar, reutilizar y modificar las partes que componen dicha solución. El utilizar n-capas también puede ser útil para acomodar diferentes tecnologías o diferentes organizaciones de negocio independizando los procesos de negocios de la base de datos y de la interfaz del usuario.

Al diseñar, desarrollar e implementar sistemas Cliente/Servidor de n-capas éstas deben ir adaptándose a las necesidades y funciones de la empresa o cliente. En este sentido se buscó crear un modelo tal que nuestro cliente pueda adaptarse rápidamente a los cambios futuros como puede ser el advenimiento de tecnologías más avanzadas o a nuevas necesidades del dominio de la aplicación.

Con el conocimiento del correcto manejo de la información y uso de últimas tecnologías y estándares se desarrolló una aplicación Web útil que obedece a necesidades específicas y definidas y que es posible escalar para cumplir con requerimientos futuros.

Internet continúa madurando hacia la plataforma principal de la computación, los sitios Web internos así como los externos están requiriendo mejores herramientas de desarrollo que estén enfocadas a agregar sofisticación y complejidad a las aplicaciones Web que las compañías construyen como soluciones de capa media capaces de conectarse con otros sistemas; estas herramientas se volverán críticas conforme crezcan los sitios Web. Actualmente existen diversas plataformas que ofrecen herramientas diversas para el desarrollo de soluciones Web, cada una con sus ventajas y desventajas y este panorama no cambiará mucho en un futuro cercano, surgirán nuevas plataformas que vendrán a reemplazar a las ya existentes, sin embargo es necesario analizar cuidadosamente el Costo/Beneficio de migración a la nueva plataforma tratando de descubrir los costos ocultos. No es simplemente instalar el *Framework* ó instalar la última versión del sistema operativo, implica un cambio en la mentalidad de los desarrolladores que están acostumbrados a cierta plataforma. El cambio implica capacitación y predisposición al cambio.

De acuerdo a la experiencia al momento de desarrollar la presente solución determinamos que nuestra solución tendrá un tiempo de vida aproximado de 3 a 4 años sin tener un cambio significativo y con el correspondiente mantenimiento, esto no quiere decir que la aplicación dejará de ser funcional, simplemente quiere decir que en ese período dada la tendencia observada en la historia de los sistemas computacionales, nuestra aplicación aunque funcional puede verse obsoleta comparada con los sistemas de ese momento. Una vez cumplido este período o antes en caso de ser necesario podría hacerse una migración a otro tipo de plataforma tal vez no Windows. Estimamos que tal migración tendrá un costo aproximado de entre el 40 y el 60% del costo total de desarrollo original, la mayor parte de estos costos está relacionada con la capacitación de los recursos humanos. Esto debido a que no sólo se cambia la tecnología de base para los desarrollos sino que además cambian

los lenguajes de programación y el modelo de objetos al cual se está acostumbrado. Por otro lado nuestra solución es una buena base para un futuro cambio, tomando en cuenta que no existía ninguna aplicación precedente para este sistema y tuvo que diseñarse la aplicación partiendo únicamente de la información obtenida y los requerimientos del cliente, por lo que esta solución será la base para cualquier desarrollo o migración posterior. Como ya mencionamos antes el haber utilizado la arquitectura de n-capas facilitará aún más tal migración ya que si se requiere hacer alguna modificación en el código sólo se necesita hacer el cambio en un solo lugar, ya sea en el almacén de datos, la capa de presentación o en las reglas de negocios, sin afectar drásticamente a las demás capas que tienen una considerable interacción, estas pueden verse como áreas funcionales distintas. Por ejemplo, un diseñador gráfico puede realizar modificaciones al diseño sin que cambie absolutamente nada del programa ya que el contenido HTML esta separado del código.

Queda claro que cada día más y más personas se conectan a Internet debido al deseo de estar comunicados de manera global y para lograr esta conexión cada día surgen más y más equipos disímiles (*Hand helds*, computadoras personales, servidores, *tablet pcs*, teléfonos celulares, etc.) y nuevas tecnologías para conectarse cada vez con un mejor ancho de banda. Por lo que al desarrollar cualquier aplicación Web esta deberá ser capaz de adaptarse a las necesidades del cliente y esto solamente es posible actualmente (y lo seguirá siendo en un futuro) si se utilizan estándares que permitan la intercomunicación entre aplicaciones y componentes e incluso entre plataformas distintas. Es por ello que al diseñar nuestra solución Web se escogieron tecnologías que cumplieran con estándares. A continuación se muestran algunas de las tecnologías utilizadas en la solución Web propuesta junto con su justificación, por último explicamos un poco algunas mejoras que podrían añadirse a la solución para darle mayor funcionalidad y hacerla más atractiva y competitiva a futuro. Por supuesto estas mejoras no fueron implementadas debido a que estaban fuera del alcance del proyecto, sin embargo consideramos que en un futuro si existe el interés por parte del cliente podrían llegar a implementarse.

## 5.2 TECNOLOGÍA .NET

Como propuesta de integración futura, la tecnología .NET ya es una herramienta empleada, se utilizó para la construcción de la solución, en esta breve descripción se presentan los beneficios a futuro del empleo de esta tecnología.

.Net es una no es solo una nueva tecnología, conlleva una filosofía con una visión a futuro, consiste en una plataforma de software que conecta información, sistemas, servicios, personas y dispositivos en una época donde la importancia de la comunicación y de la información con el apoyo de Internet, permite el acceder a recursos de forma casi ilimitada y estar siempre informados. Actualmente ya no solo se busca compartir información mediante una computadora sino el obtener un valor agregado de la misma, dado el constante aumento de nuevos dispositivos personales que buscan adquirir información y compartirla mediante Internet, estos requieren de la implementación de nuevas tendencias informáticas y de servicios que sean adecuados a las nuevas necesidades.

Es así como .Net pretende marcar el fin de una era y el inicio de otra en cuanto al cómputo, la plataforma .NET conecta una gran variedad de tecnologías de uso personal y de negocios, de teléfonos celulares a servidores corporativos, permitiendo el acceso a información importante, donde y cuando se necesiten. Microsoft está colaborando con otros líderes de la industria que comparten una visión de tecnología para trabajar juntos de una forma transparente. El reto es hacer que los dispositivos de computación trabajen juntos de una forma confiable y automática, brindando la información y recursos que se necesiten.

La forma como se busca este fin es con el desarrollo de estándares de software abierto que harán posible que todos los tipos de computadoras compartan información y brinden servicios personalizados por Internet, a cualquier dispositivo que esté conectado. Como la ha demostrado la historia, la computación ha cambiado al mundo y así como el teléfono cambió la forma de comunicarse, las computadoras serán tan fáciles de usar que la gente difícilmente se dará cuenta de ellas, tan familiares y eficientes que en efecto se volverán invisibles.

Al utilizar la tecnología .NET en esta solución, permitirá que el sistema y su aplicación sean consumidos por cualquier tipo de cliente, independientemente del sistema operativo del mismo, del tipo de computadora o dispositivo móvil que se utilice, incluso del lenguaje de programación utilizado para desarrollarlo. Esto es y será posible debido a que la plataforma .NET está basada en Servicios Web que a su vez están basados en estándares, de forma que todos los productos estarán conectados a través de Servicios Web XML con la capacidad de intercambiar datos de forma rápida, económica y segura.

El desarrollar bajo esta plataforma y apostando al futuro permitirá que otras aplicaciones desarrolladas bajo este mismo esquema puedan intercambiar servicios y transacciones con nuestra aplicación para que así nuestra solución concentrada en un servidor no se encuentre sólo como un dispositivo aislado sino que forme parte una red de dispositivos trabajando conjuntamente, ofreciendo de esta forma soluciones más amplias y ricas en contenido, lo cual beneficiara a los usuarios, la Secretaria del Medio Ambiente podrá ofrecer sus servicios a los clientes, en el momento correcto y de forma precisa.

### 5.2.1 SERVICIOS WEB

Los Servicios Web son la más innovadora tecnología para los negocios y aplicaciones en la Web. Los Servicios Web XML utilizan tecnologías programables y reutilizables que aprovechan la flexibilidad de Internet. Con ellos es posible tener una infinidad de aplicaciones conectados en red, ya sea que se ejecuten en diferentes plataformas, proporcionando información a todo mundo, los Servicios Web tienen como base un conjunto de estándares abiertos, incluyendo XML, SOAP, WSDL y UDDI, los cuales son controlados por el "*World Wide Web Consortium*" (W3C). Trabajar con .NET significa usar protocolos abiertos que unen sistemas y aplicaciones existentes, permitiendo aprovechar mejor todos los beneficios que ofrecen. La interoperabilidad total a través de los Servicios Web XML será una realidad, si se garantiza que la interpretación e implementación de las especificaciones de los Servicios Web sean consistentes y de acuerdo común entre todas las empresas de tecnología, es así como empresas líderes de los sectores de software y tecnología, tienen el compromiso de promover la interoperabilidad entre los Servicios Web

XML, con base en definiciones comunes aceptadas por el mercado y que apoyan los estándares XML, que ayudará a que el avance de los Servicios Web sea de manera estructurada y coherente.

Los servicios Web son aplicaciones reutilizables escritas en diversos lenguajes, con una compilación que entrega lenguaje XML, un lenguaje universal escrito para el intercambio de datos. Tales servicios permiten el intercambio de datos utilizando la red Internet o incluso dentro de una Intranet entre dispositivos o fuentes que pueden actuar como recipientes o consumidores de estos servicios (ver figura 5-1), algunos de estos escenarios pueden ser:

- Cliente a cliente.- Clientes o dispositivos inteligentes pueden contener o consumir servicios Web para intercambiar datos en cualquier momento o en cualquier lugar.
- Cliente a servidor.- Los servicios Web pueden compartir información desde un servidor de aplicaciones hacia una PC o algún dispositivo móvil vía Internet.
- Servidor a servidor.- Los servicios Web XML proporcionan una interfaz común entre aplicaciones existentes dentro de un ambiente de servidores independientes.
- Servicio a servicio.- Los servicios Web XML pueden incluso trabajar juntos para crear una operación más compleja de los datos.

El usar servicios Web en nuestra solución permitirá que otras aplicaciones invoquen los servicios que ofrecemos y esto sin importar o sin que a tales aplicaciones les afecte el lenguaje utilizado para el desarrollo de los mismos e independientemente del dispositivo que acceda a tal servicio.

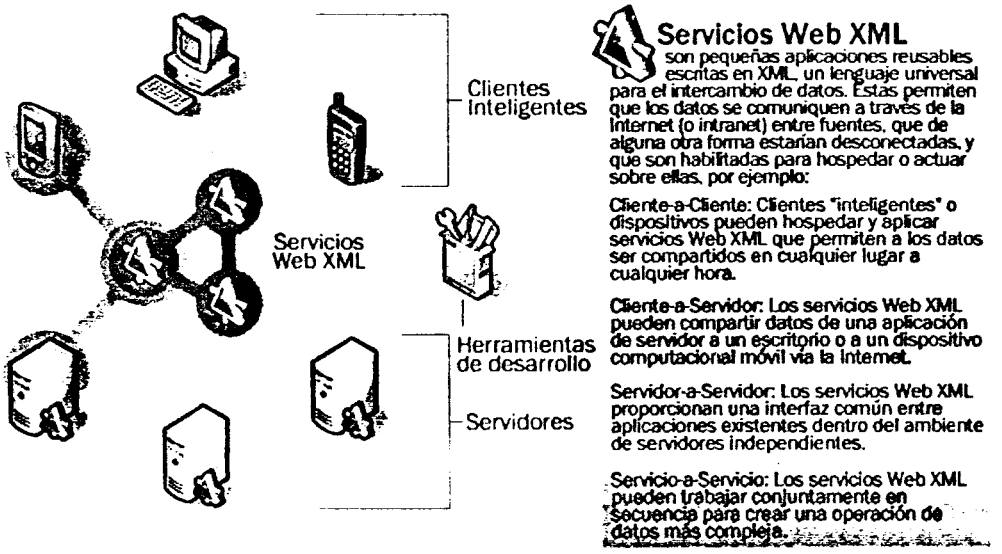



**Figura 5-1 Descripción y medio ambiente de los servicios Web**

### 5.2.2 LENGUAJE XML

La solución Web al contener lenguaje XML incrustado en el código, estará preparada para el software de la próxima generación.

HTML resultó adecuado para el nacimiento de Internet, pero Internet se ha convertido en un centro para la información y los negocios, ha evolucionado y HTML ya no es capaz de satisfacer completamente sus necesidades. El fracaso de los navegadores de Internet a la hora de cumplir los estándares de HTML al volverse dependientes de "plug-ins", la dificultad para validar los documentos HTML, un sistema de vínculos pobre y falta de respaldo internacional han convertido a HTML en una elección muy limitada para el futuro. SGML es una herramienta excelente y de gran potencia, capaz de documentar sistemas complejos, pero, desgraciadamente, es demasiado compleja para las necesidades actuales de Internet.

XML es ideal para la próxima generación de aplicaciones de Internet por ejemplo para el comercio electrónico (*e-commerce*). XML es un lenguaje de marcado más ligero y sencillo, flexible, fácil de usar y que se puede utilizar para documentos internacionales, además es ideal para almacenar datos y para enviar mensajes, y los documentos XML se pueden validar. En pocos años XML se refinará hasta convertirse en una herramienta increíblemente poderosa que originará la siguiente evolución de Internet.

Está demostrado que el desarrollo de aplicaciones es más fácil, más rápido y más barato al utilizar el entorno de desarrollo .NET (.NET *Framework*) y la herramienta de desarrollo Visual Studio .NET, como lo pudimos comprobar al momento de desarrollar la presente aplicación. El utilizar visual Studio .NET nos permite desarrollar en un ambiente multilenguaje. .NET contiene gran cantidad de código en las librerías de clase lo cual conlleva en un ahorro en la codificación, incrementando la productividad.

Trabajar con .NET permitió trabajar con un caché basado en XML de los datos de la base de datos de producción en vez de trabajar directamente con ella, lo cual libera conexiones con la base de datos dando como resultado una mayor escalabilidad.

### 5.2.3 SERVIDOR DE BASE DE DATOS

La base de datos fue normalizada para que esta mantuviera la integridad y no repercutiera en el rendimiento de la misma, el modelo planteado es totalmente funcional y cubre perfectamente a las necesidades de nuestro cliente, sin embargo en un futuro podría replantearse un modelo distinto al actual. El modelo actual permite la inserción de nuevos campos en cada una de las tablas en caso de ser necesario, es decir es un modelo que ofrece flexibilidad, o podría replantearse el modelo completamente, con la seguridad de que los datos estarán correctamente almacenados y un modelo nuevo solo tendría que importar dichos datos hacia el nuevo modelo. Es así que nuestra aplicación irá adaptándose a medida que cambien las necesidades del cliente, en este caso la Secretaría del Medio Ambiente.

Al haber sido desarrollada y administrada la base de datos usando SQL Server 2000 se simplificarán las labores de mantenimiento de la base de datos dada la facilidad de uso y la

interfaz intuitiva; como ejemplo, el realizar copias de seguridad de la base de datos puede automatizarse cada determinado tiempo por lo general en un horario que no afecte el rendimiento del sistema, ya que la aplicación al ser una solución consultada por Internet debe estar en funcionamiento los 7 días de la semana las 24 horas del día es decir la misma debe tener una alta disponibilidad, de tal forma que podría escogerse un horario no tan crítico para la aplicación, por ejemplo a las 2 de la mañana, programando una tarea en SQL Server 2000 puede realizarse esta tarea sin que ni siquiera estemos presentes.

## 5.3 CONTENIDO MULTIMEDIA

La aplicación fue desarrollada pensando que en un futuro se podrían introducir nuevos datos a la base de datos y esta deberá responder de la misma forma y crecer junto con las necesidades del cliente, por otro lado los datos estarán disponibles en cada momento y en cualquier lugar para su consulta de forma automatizada gracias a la interfaz Web disponible por Internet. Algunos de los datos que podrían agregarse para darle una mayor funcionalidad a la solución Web planteada sería el agregar material multimedia, por ejemplo videos, los cuales podrían estar almacenados en una base de datos paralela manteniendo cierta independencia o incluso podrían agregarse como parte de un parámetro más de la misma base de datos, ya que por ejemplo se podría ver un video que mostrara la forma en como se realizó el muestreo, la consulta a tal material puede restringirse empleando los derechos digitales (DRM, *Digital Rights Management*) sobre el mismo material y/o transmitiendo el material en vivo usando *streaming* a través de Internet. No cabe duda que la transmisión de contenido multimedia a través de Internet será cada vez más importante. La tecnología de *streaming* es un mercado con futuro y grandes compañías ya están luchando por el mercado. La velocidad de conexión a Internet aumentará con el tiempo y con ella aumentará la calidad de las transmisiones.

La tecnología de *streaming* se utiliza para agilizar la descarga y ejecución de audio y vídeo en Internet, ya que permite escuchar y visualizar los archivos mientras se están descargando, Si no se utiliza *streaming*, para mostrar un contenido multimedia en Internet, es necesario descargar primero el archivo entero en la computadora cliente y después ejecutarlo para poder ver y oír lo que el archivo contiene. Sin embargo, el *streaming* permite que esta tarea se realice de una manera más rápida y pueda verse y escucharse el contenido multimedia durante la descarga.

El *streaming* funciona de la siguiente manera: El cliente se conecta con el servidor y éste le empieza a mandar el archivo, el cliente comienza a recibir el archivo y construye un buffer donde empieza a guardar la información. Cuando se ha llenado el buffer con una pequeña parte del archivo, el cliente lo empieza a mostrar y continúa con la descarga. El sistema está sincronizado para que el archivo se pueda ver mientras que el archivo se descarga, de modo que cuando el archivo acaba de descargarse también termina de visualizarse. Si en algún momento la conexión sufre descensos de velocidad se utiliza la información que hay en el buffer, de modo que la transmisión se mantiene de manera fluida y este descenso de velocidad pasa casi desapercibido para el cliente. Si la comunicación se corta demasiado tiempo o la velocidad de transmisión es demasiado lenta, el buffer se vacía y la ejecución del archivo terminaría también hasta que se restaure la señal.

Otra tecnología de cómputo que haría posible extender los servicios de nuestra solución solo a un alcance mayor sino tener una disponibilidad de mayor rendimiento, es el caso del empleo de balanceo de carga de aplicaciones y el cómputo distribuido. Permitiendo que las soluciones y servicios estén siempre disponibles, de forma segura y confiable con un mínimo de inversión, ya que la implementación de estos sistemas no requiere de grandes cambios en el proyecto original, debido a que cuenta con un alto grado de portabilidad entre equipos servidores y clientes.

## 5.4 SISTEMAS DE INFORMACIÓN GEOGRÁFICA

Con el empleo de los sistemas GIS (Sistema de Información Geográfica), se podría rediseñar completamente la interfase de navegación de lugares de muestreo. Los sistemas "GIS" son una herramienta computacional que permite integrar información referenciada espacialmente con bases de datos, proporcionando de forma cómoda información en una plataforma con alto impacto visual y gran capacidad de síntesis informativa.

Un ejemplo de este tipo de sistema y que podría integrarse a la actual solución sería ArcExplorer con el cual podría manipularse la información contenida en la base de datos para ser mostrada en un mapa de navegación, donde queden registrados los lugares de muestreo y al seleccionar alguna zona en el mapa y específicamente al seleccionar algún lugar este refleje la información de caracterización del lugar. Mostrando la información completa obtenida de las muestras tomadas en el lugar. Usando dicha aplicación la información sería consultada directamente de Internet de la solución desarrollada, para ser mostrada localmente en cualquier equipo en cualquier parte del mundo, solo sería necesario instalar la aplicación ArcExplorer para tener acceso de manera gráfica por medio de mapas a la información requerida.

Por otro lado, ArcExplorer permitiría distribuir la información de este proyecto de manera fácil ya que en un CD podría transportarse los datos así como una aplicación GIS instalable para la visualización y consulta de los mismos.

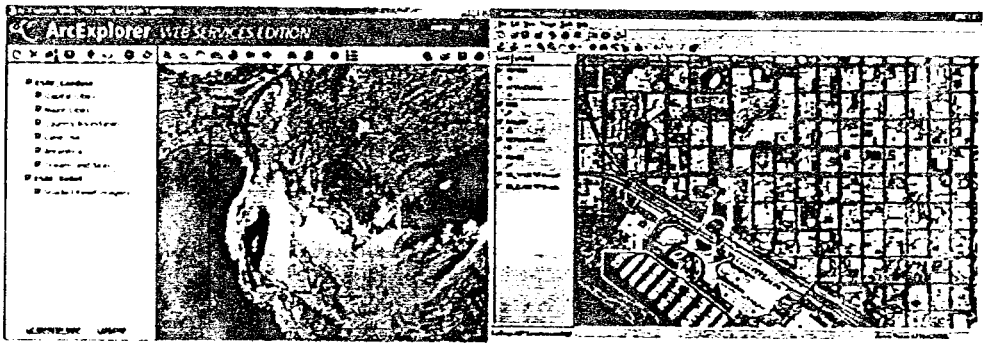



**Figura 5-2 Ejemplos del empleo de ARCEXPLORER**

# CAPÍTULO 6 CONCLUSIONES

## 6.1 CONCLUSIONES

El desarrollo de la presente tesis refleja la recopilación y aplicación de los conocimientos que hemos adquirido durante nuestra vida académica, especialmente de los años de estudio dentro de la Facultad de Ingeniería que nos dio una formación a nuestro parecer bastante integral. Sin embargo, el presente trabajo no lo vemos como el final o la conclusión de nuestra formación sino como el principio como profesionistas y del impulso a la investigación de nuevas técnicas, metodologías, aplicación de la ingeniería de software o demás temas afines a nuestra carrera y que son de interés para la sociedad, aplicando todo lo anterior a cualquier tipo de problema que sea factible de ser solucionado por medios computacionales.

Debido a la rápida evolución del cómputo e informática, con el apoyo de las nuevas herramientas y tendencias informáticas logramos desarrollar de forma confiable y segura una aplicación que no sólo cumple con las expectativas del cliente sino que aportamos el inicio para la difusión de información de una forma simplificada y disponible por medios económicos, como es el uso de Internet, tomando un nuevo enfoque para futuras aplicaciones y adaptaciones que constantemente se presentan hoy en día.

Al terminar nuestra carrera comenzamos a laborar como becarios en el Instituto de Ingeniería lo cual nos ha ayudado a adquirir en cierta manera la experiencia laboral que complementa la experiencia adquirida dentro de las aulas de la Facultad al enfrentarnos con problemas reales que hemos tenido que resolver aplicando nuestros conocimientos y es precisamente de esta manera como nos encontramos con el Proyecto del "PROGRAMA DE MANEJO AMBIENTAL DE LOS RESIDUOS PROVENIENTES DE DESAZOLVE DEL SISTEMA DE DRENAJE DEL DF Y DE LAS PRESAS, ASÍ COMO DE PLANTAS DE TRATAMIENTO DE AGUAS RESIDUALES MUNICIPALES" programa establecido entre la Secretaría del Medio Ambiente del Distrito Federal y el Grupo de Tratamiento y Reuso del Instituto de Ingeniería de la UNAM, del cual una parte era el presentar un banco de información y difusión de datos que mostrara los resultados obtenidos en la investigación realizada por el grupo, es ahí donde surge la inquietud de presentar tal parte del proyecto como nuestro proyecto de tesis; nuestra propuesta fue el realizar tal banco de información y difusión de datos como una solución Web para lo cual tuvimos que realizar todo el proceso de ingeniería de software, desde recolectar información, diseñar la aplicación, la base de datos a la cual se conecta la aplicación y la cual guarda la información recopilada y diseñar la interfaz Web que mostrara tales datos de forma accesible.

La idea de desarrollar una solución Web para el problema original responde a la necesidad de evitar el almacenamiento inútil de extensos informes en papel o formato digital, y en vez de ello el presentar los datos en una página de Internet en una forma dinámica y fácil de comprender la información, utilizándose para ello un modelo basado en Web.

En el presente proyecto se evaluaron diversos parámetros: físicos, químicos y microbiológicos que sirvieron para clasificar los materiales y establecer criterios para su manejo y destino final, tales datos provenientes del estudio de las diversas muestras se recolectaron en una única hoja de cálculo(una tabla de Excel), la cual si bien es una aplicación disponible en casi cualquier computadora de escritorio, su consulta no resultaba del todo práctica, ya que se contaba con una infinidad de datos con posibilidades de crecimiento a futuro y la consulta de los mismos era una

tarea tediosa al tener que revisar todos los registros y con posibilidad de confundirse con datos innecesarios. Es así como la solución desarrollada permitió reorganizar estos datos en una base de datos que divide y organiza los datos en varias tablas que permite realizar búsquedas y así revisar cientos de registros para acceder sólo a los datos necesarios y en periodos de segundos de forma transparente para el usuario y de una forma segura para los datos al mantenerse la integridad de los mismos y permitiendo al usuario acceder únicamente a la información para la cual tiene los permisos necesarios.

Al utilizar una presentación mediante páginas de Internet los datos presentados al usuario que consulte la base de datos ofrecen una interfaz amigable al usuario y de forma ordenada. La mayor ventaja para el usuario final al haber sido desarrollada una solución Web es el poder realizar la consulta de los datos desde cualquier computadora en cualquier parte del mundo siempre y cuando se cuente con una conexión a Internet, hecho imposible mediante el método anterior en una hoja de cálculo ya que para poder intercambiar la información sería necesario transferir y portar la información en un archivo con la correspondiente incomodidad y falta de seguridad para los datos.

La ventaja de utilizar una aplicación Web dividida "en capas" nos brindó la ventaja de aislar las reglas de negocios, los datos y la presentación de los datos de modo tal que para cualquier cambio eventual sólo se deberá tocar un módulo específico, así como al momento de plantear la escalabilidad de nuestro sistema no debamos afrontar grandes modificaciones ya que al factorizar una aplicación en partes lógicas resulta más fácil de generar, reutilizar y modificar, por otro lado permite acomodar diferentes tecnologías dentro de una solución como un todo. De esta forma esta solución seguirá respondiendo a las necesidades de nuestro cliente con el paso del tiempo.

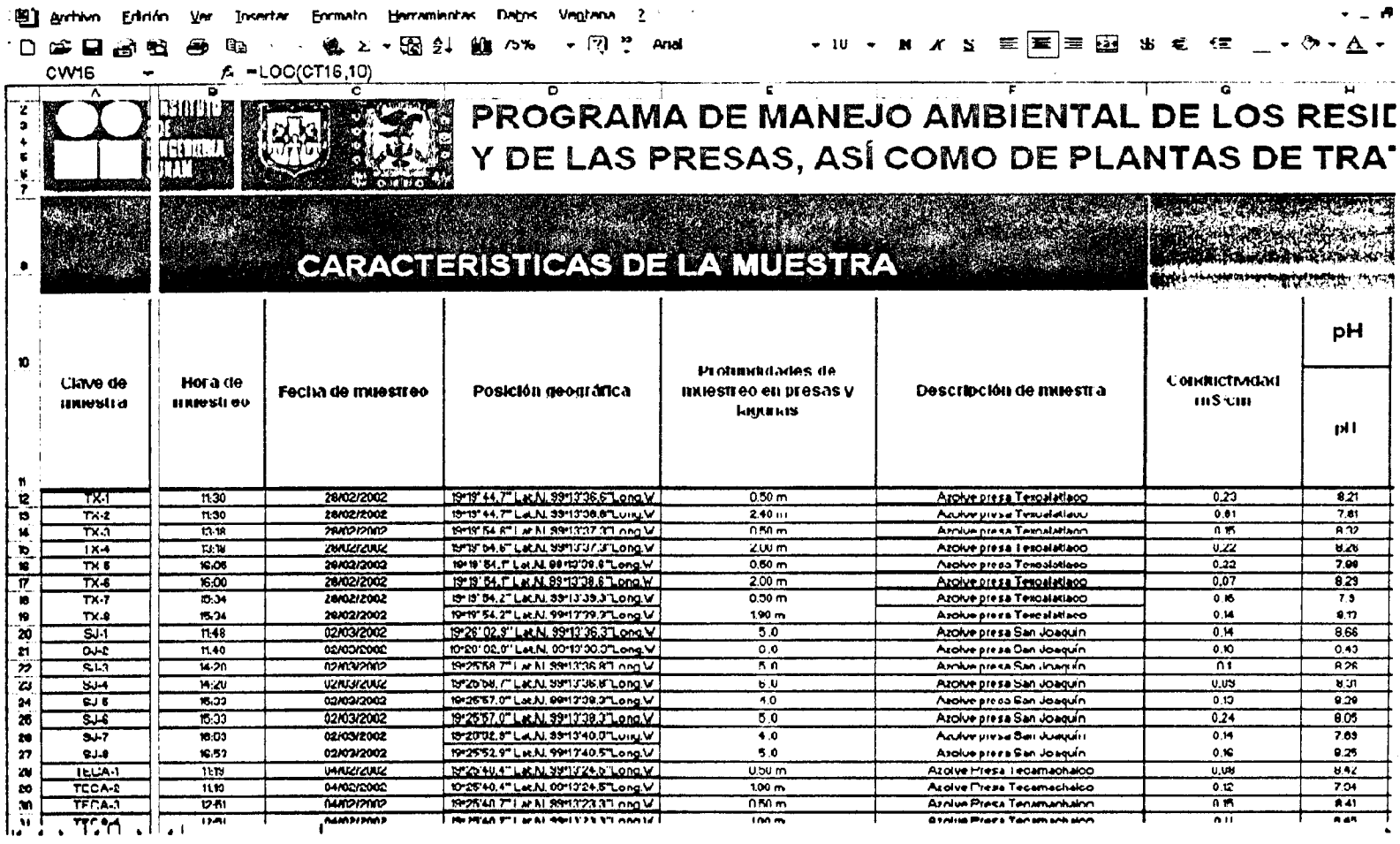
En cuanto al impacto social que tendrá nuestra solución, no solo se refiere al manejo de datos, control y su disponibilidad, ya que no sólo recurre a los clientes finales, sino que permite tener a una sociedad informada de las actividades, tareas, uso de recursos, necesidades que se encuentran a su alrededor. Siendo este punto muy importante para el apoyo en las tareas de administración, control y empleo adecuado de recursos con los que cuenta una institución, empresa o en este caso el gobierno.

La magnitud de este proyecto fue tal que nos permitió ampliar nuestros conocimientos técnicos y en nuestra formación como ingenieros, para poder obtener una solución al problema planteado, ya que obtuvimos la experiencia de poder interactuar con un cliente para llevar a cabo una solución que cumpliera con las expectativas y necesidades del cliente, en base a la limitación de sus recursos, preferencias y necesidades teniendo que llevarse a cabo pruebas exhaustivas que dio pie a revisar continuamente la aplicación hasta obtener un resultado satisfactorio para el cliente. Por lo tanto nos muestra la necesidad de estar constantemente actualizados dentro del ambiente computación, dentro de los aspectos modernos de comunicación, desarrollo de soluciones, herramientas y de la perspectiva de nuevas técnicas para atacar nuevos problemas que se presentan constantemente.

El desarrollo de este proyecto puede ser la base para la solución en futuros proyectos relacionados con esta problemática ambiental en ciudades como Guadalajara, Monterrey y Ciudad Juárez, que presentan problemas similares.

# ANEXOS

## Recolección de datos de muestreo



Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

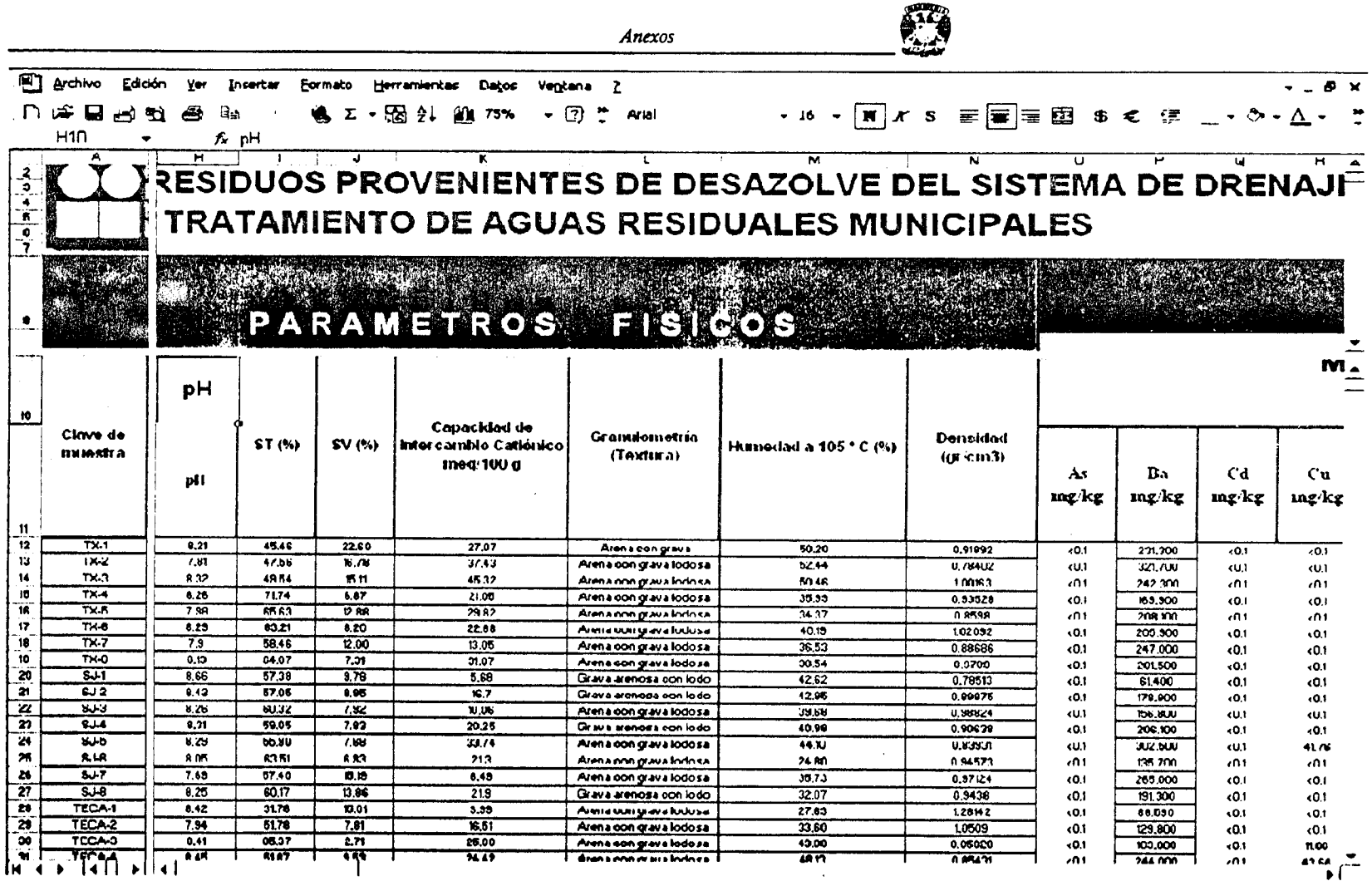
CW16 =LOC(CT16,10)

PROGRAMA DE MANEJO AMBIENTAL DE LOS RESID
Y DE LAS PRESAS, ASÍ COMO DE PLANTAS DE TRA

CARACTERISTICAS DE LA MUESTRA

| Clave de muestra | Hora de muestreo | Fecha de muestreo | Posición geográfica | Profundidades de muestreo en presas y lagunas | Descripción de muestra | Conductividad mS/cm | pH pH |
|---|---|---|---|---|---|---|---|
| TX-1 | 11:30 | 28/02/2002 | 19°19' 44.7" Lat.N. 99°13'36.6" Long.W | 0.50 m | Azolve presa Tecolalatlaco | 0.23 | 8.21 |
| TX-2 | 11:30 | 28/02/2002 | 19°19' 44.7" Lat.N. 99°13'36.6" Long.W | 2.40 m | Azolve presa Tecolalatlaco | 0.81 | 7.81 |
| TX-3 | 13:18 | 28/02/2002 | 19°19' 54.6" Lat.N. 99°13'37.3" Long.W | 0.50 m | Azolve presa Tecolalatlaco | 0.15 | 8.12 |
| TX-4 | 13:18 | 28/02/2002 | 19°19' 54.6" Lat.N. 99°13'37.3" Long.W | 2.00 m | Azolve presa Tecolalatlaco | 0.22 | 8.26 |
| TX-5 | 16:06 | 28/02/2002 | 19°19' 54.1" Lat.N. 99°13'39.6" Long.W | 0.50 m | Azolve presa Tecolalatlaco | 0.22 | 7.99 |
| TX-6 | 16:00 | 28/02/2002 | 19°19' 54.1" Lat.N. 99°13'39.6" Long.W | 2.00 m | Azolve presa Tecolalatlaco | 0.07 | 8.29 |
| TX-7 | 15:34 | 28/02/2002 | 19°19' 54.2" Lat.N. 99°13'39.3" Long.W | 0.50 m | Azolve presa Tecolalatlaco | 0.16 | 7.9 |
| TX-8 | 15:34 | 28/02/2002 | 19°19' 54.2" Lat.N. 99°13'39.3" Long.W | 1.90 m | Azolve presa Tecolalatlaco | 0.14 | 8.17 |
| SJ-1 | 11:48 | 02/03/2002 | 19°26' 02.9" Lat.N. 99°13'36.3" Long.W | 5.0 | Azolve presa San Joaquín | 0.14 | 8.66 |
| SJ-2 | 11:40 | 02/03/2002 | 19°26' 02.9" Lat.N. 99°13'36.3" Long.W | 0.0 | Azolve presa San Joaquín | 0.10 | 0.40 |
| SJ-3 | 14:20 | 02/03/2002 | 19°25'58.7" Lat.N. 99°13'36.8" Long.W | 5.0 | Azolve presa San Joaquín | 0.1 | 8.26 |
| SJ-4 | 14:20 | 02/03/2002 | 19°25'58.7" Lat.N. 99°13'36.8" Long.W | 5.0 | Azolve presa San Joaquín | 0.09 | 8.31 |
| SJ-5 | 16:00 | 02/03/2002 | 19°25'57.0" Lat.N. 99°13'38.3" Long.W | 4.0 | Azolve presa San Joaquín | 0.13 | 9.29 |
| SJ-6 | 16:00 | 02/03/2002 | 19°25'57.0" Lat.N. 99°13'38.3" Long.W | 5.0 | Azolve presa San Joaquín | 0.24 | 8.05 |
| SJ-7 | 16:03 | 02/03/2002 | 19°25'52.8" Lat.N. 99°13'40.0" Long.W | 4.0 | Azolve presa San Joaquín | 0.14 | 7.69 |
| SJ-8 | 16:53 | 02/03/2002 | 19°25'52.9" Lat.N. 99°13'40.5" Long.W | 5.0 | Azolve presa San Joaquín | 0.16 | 9.25 |
| TECA-1 | 11:19 | 04/02/2002 | 19°25'40.4" Lat.N. 99°13'24.5" Long.W | 0.50 m | Azolve Presa Tecamachalco | 0.09 | 8.42 |
| TECA-2 | 11:19 | 04/02/2002 | 19°25'40.4" Lat.N. 99°13'24.5" Long.W | 1.90 m | Azolve Presa Tecamachalco | 0.12 | 7.04 |
| TECA-3 | 12:51 | 04/02/2002 | 19°25'40.7" Lat.N. 99°13'23.3" Long.W | 0.50 m | Azolve Presa Tecamachalco | 0.15 | 8.41 |
| TECA-4 | 12:51 | 04/02/2002 | 19°25'40.7" Lat.N. 99°13'23.3" Long.W | 1.90 m | Azolve Presa Tecamachalco | 0.11 | 8.45 |

**Conjunto original de datos**



Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

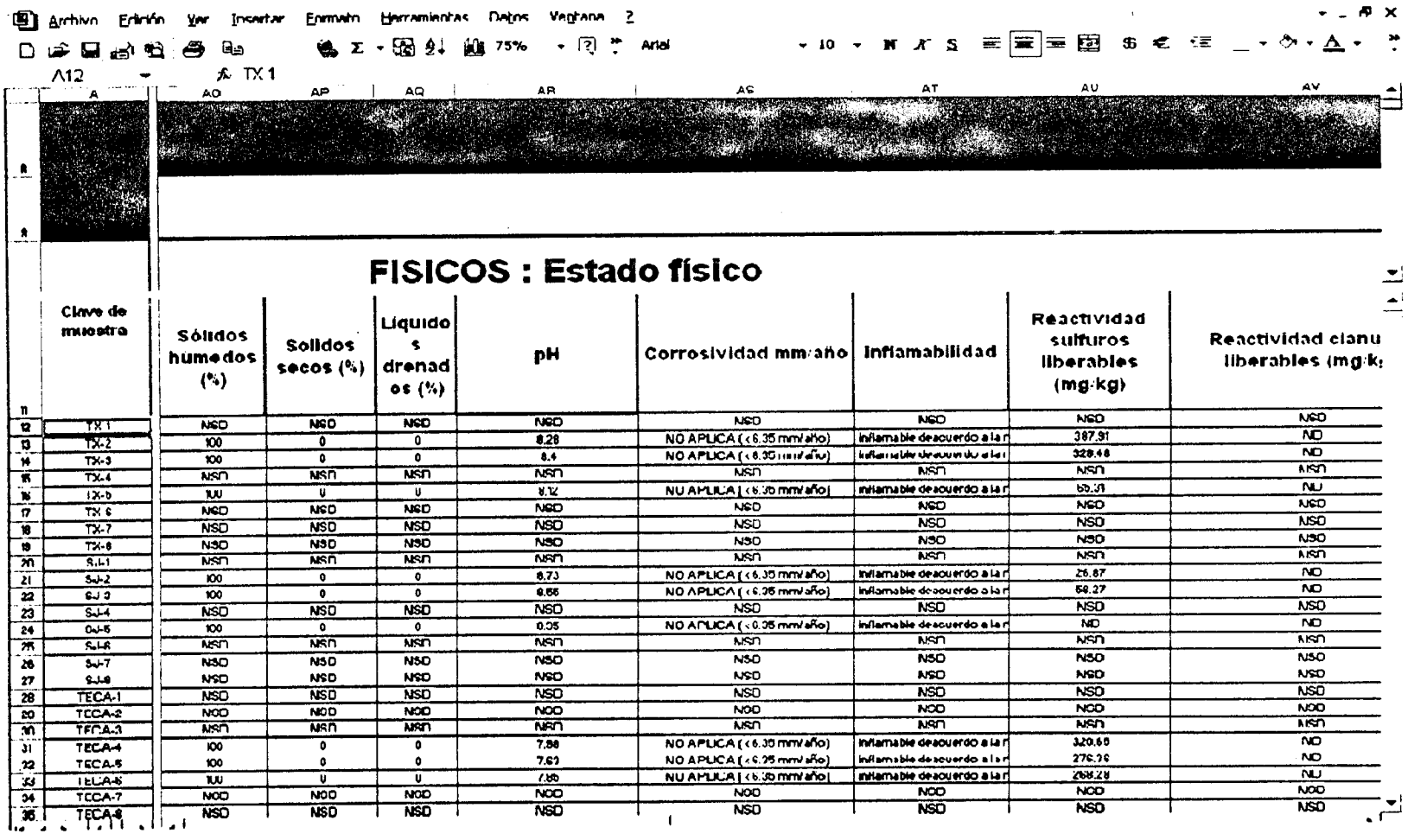
H10 pH

RESIDUOS PROVENIENTES DE DESAZOLVE DEL SISTEMA DE DRENAJI
TRATAMIENTO DE AGUAS RESIDUALES MUNICIPALES

PARAMETROS FISICOS

| Clave de muestra | pH pH | ST (%) | SV (%) | Capacidad de Intercambio Catiónico meq/100 g | Granulometría (Textura) | Humedad a 105 ° C (%) | Densidad (g/cm3) | As mg/kg | Ba mg/kg | Cd mg/kg | Cu mg/kg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TX-1 | 9.21 | 45.46 | 22.60 | 27.07 | Arena con grava | 50.20 | 0.91992 | <0.1 | 221,200 | <0.1 | <0.1 |
| TX-2 | 7.81 | 47.56 | 16.78 | 37.43 | Arena con grava lodosa | 52.44 | 0.784U2 | <0.1 | 321,700 | <0.1 | <0.1 |
| TX-3 | 8.32 | 49.84 | 15.11 | 45.32 | Arena con grava lodosa | 50.46 | 1.00163 | <0.1 | 242,300 | <0.1 | <0.1 |
| TX-4 | 8.26 | 71.74 | 6.87 | 21.06 | Arena con grava lodosa | 35.93 | 0.93528 | <0.1 | 169,900 | <0.1 | <0.1 |
| TX-5 | 7.98 | 85.63 | 12.88 | 29.82 | Arena con grava lodosa | 34.17 | 0.8598 | <0.1 | 208,100 | <0.1 | <0.1 |
| TX-6 | 8.29 | 80.21 | 8.20 | 22.88 | Arena con grava lodosa | 40.19 | 1.02092 | <0.1 | 209,900 | <0.1 | <0.1 |
| TX-7 | 7.9 | 58.46 | 12.00 | 13.05 | Arena con grava lodosa | 36.53 | 0.88686 | <0.1 | 247,000 | <0.1 | <0.1 |
| TX-0 | 0.19 | 04.07 | 7.01 | 31.07 | Arena con grava lodosa | 30.54 | 0.9700 | <0.1 | 201,500 | <0.1 | <0.1 |
| SJ-1 | 8.66 | 57.38 | 9.78 | 5.68 | Grava arenosa con lodo | 42.62 | 0.78513 | <0.1 | 61,400 | <0.1 | <0.1 |
| SJ-2 | 9.43 | 87.06 | 8.96 | 16.7 | Grava arenosa con lodo | 42.96 | 0.999/6 | <0.1 | 179,900 | <0.1 | <0.1 |
| SJ-3 | 8.26 | 80.32 | 7.52 | 10.06 | Arena con grava lodosa | 39.59 | 0.98824 | <0.1 | 156,800 | <0.1 | <0.1 |
| SJ-4 | 8.21 | 59.05 | 7.92 | 20.25 | Grava arenosa con lodo | 40.98 | 0.90629 | <0.1 | 206,100 | <0.1 | <0.1 |
| SJ-5 | 8.29 | 66.80 | 7.88 | 33.74 | Arena con grava lodosa | 44.10 | 0.83501 | <0.1 | 302,500 | <0.1 | 41.76 |
| SJ-6 | 8.05 | 83.51 | 6.83 | 21.3 | Arena con grava lodosa | 24.88 | 0.94573 | <0.1 | 135,700 | <0.1 | <0.1 |
| SJ-7 | 7.68 | 87.40 | 10.18 | 6.48 | Arena con grava lodosa | 35.73 | 0.97124 | <0.1 | 265,000 | <0.1 | <0.1 |
| SJ-8 | 8.25 | 80.17 | 13.86 | 21.9 | Grava arenosa con lodo | 32.07 | 0.9438 | <0.1 | 191,300 | <0.1 | <0.1 |
| TECA-1 | 8.42 | 31.76 | 10.01 | 3.39 | Arena con grava lodosa | 27.83 | 1.28142 | <0.1 | 66,050 | <0.1 | <0.1 |
| TECA-2 | 7.94 | 51.78 | 7.81 | 16.51 | Arena con grava lodosa | 33.60 | 1.0509 | <0.1 | 129,800 | <0.1 | <0.1 |
| TCCA-3 | 0.41 | 06.37 | 2.71 | 25.00 | Arena con grava lodosa | 43.00 | 0.05020 | <0.1 | 100,000 | <0.1 | 11.00 |
| TECA-4 | 8.48 | 81.87 | 4.69 | 34.49 | Arena con grava lodosa | 48.12 | 0.85431 | <0.1 | 244,000 | <0.1 | 42.64 |

**Conjunto original de datos**



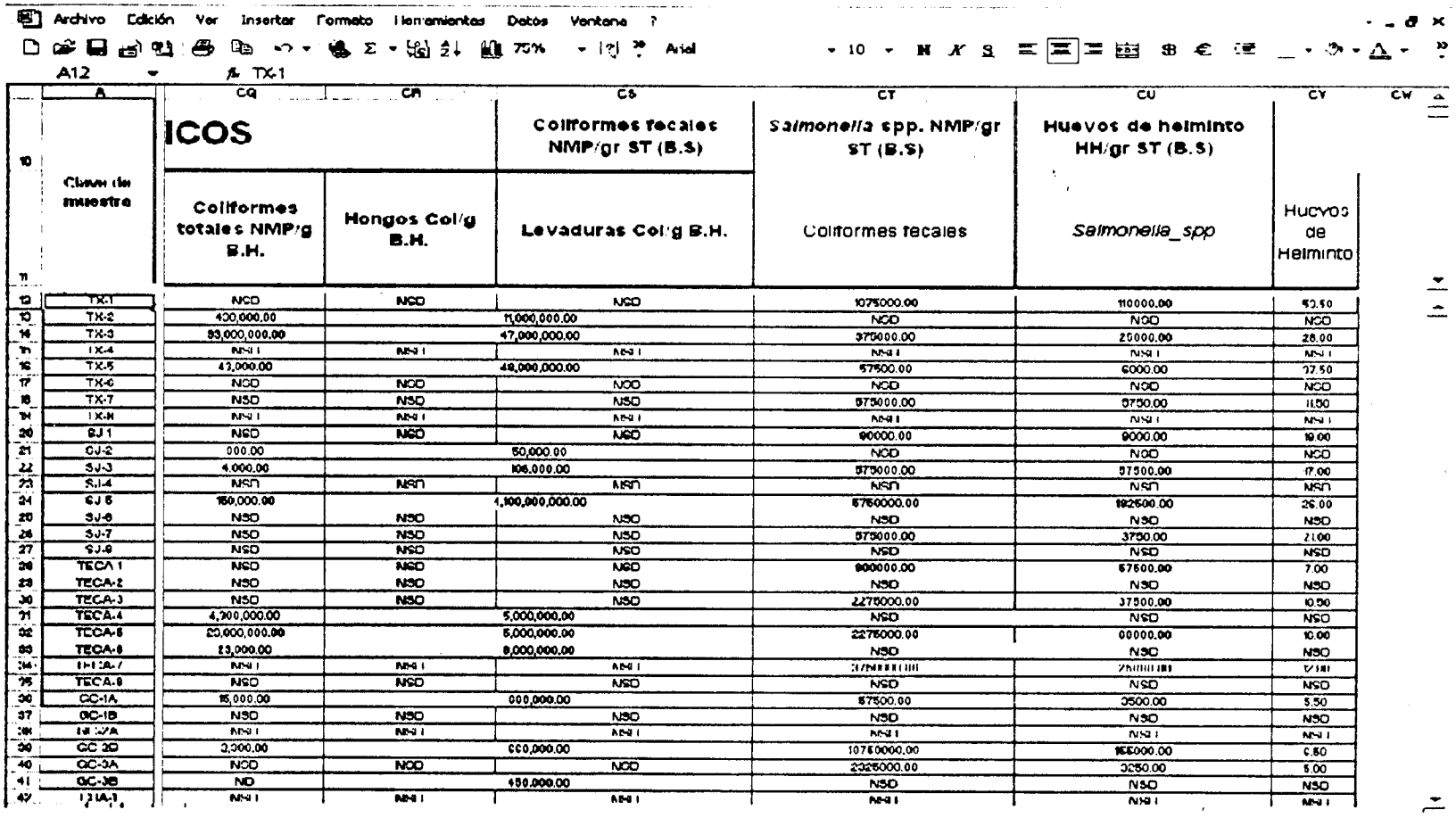
Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

A12 TX 1

## FISICOS : Estado físico

| Clave de muestra | Sólidos humedos (%) | Solidos secos (%) | Liquidos drenados (%) | pH | Corrosividad mm/año | Inflamabilidad | Reactividad sulfuros liberables (mg/kg) | Reactividad cianuros liberables (mg/k |
|---|---|---|---|---|---|---|---|---|
| TX-1 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TX-2 | 100 | 0 | 0 | 8.28 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | 387.91 | ND |
| TX-3 | 100 | 0 | 0 | 8.4 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | 328.48 | ND |
| TX-4 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TX-5 | 100 | 0 | 0 | 8.12 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | 69.31 | ND |
| TX-6 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TX-7 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TX-8 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| SJ-1 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| SJ-2 | 100 | 0 | 0 | 8.73 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | 26.87 | ND |
| SJ-3 | 100 | 0 | 0 | 8.66 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | 58.27 | ND |
| SJ-4 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| SJ-5 | 100 | 0 | 0 | 8.35 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | ND | ND |
| SJ-6 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| SJ-7 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| SJ-8 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TECA-1 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TECA-2 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TECA-3 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TECA-4 | 100 | 0 | 0 | 7.88 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | 320.65 | ND |
| TECA-5 | 100 | 0 | 0 | 7.62 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | 276.26 | ND |
| TECA-6 | 100 | 0 | 0 | 7.85 | NO APLICA (<6.35 mm/año) | Inflamable de acuerdo a la r | 268.28 | ND |
| TECA-7 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |
| TECA-8 | NSD | NSD | NSD | NSD | NSD | NSD | NSD | NSD |

**Conjunto original de datos**



Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

A12 TX-1

| | A | CQ | CR | CS | CT | CU | CV | CW |
|---|---|---|---|---|---|---|---|---|
| 10 | | ICOS | | Coliformes fecales NMP/gr ST (B.S) | *Salmonella* spp. NMP/gr ST (B.S) | Huevos de helminto HH/gr ST (B.S) | | |
| 11 | Clave de muestra | Coliformes totales NMP/g B.H. | Hongos Col/g B.H. | Levaduras Col/g B.H. | Coliformes fecales | *Salmonella_spp* | Huevos de Helminto | |
| 12 | TX-1 | NGD | NGD | NGD | 1075000.00 | 110000.00 | 50.50 | |
| 13 | TX-2 | 400,000.00 | | 11,000,000.00 | NGD | NGD | NGD | |
| 14 | TX-3 | 93,000,000.00 | | 47,000,000.00 | 375000.00 | 25000.00 | 25.00 | |
| 15 | TX-4 | NSD | NSD | NSD | NSD | NSD | NSD | |
| 16 | TX-5 | 43,000.00 | | 48,000,000.00 | 57500.00 | 6000.00 | 27.50 | |
| 17 | TX-6 | NGD | NGD | NGD | NGD | NGD | NGD | |
| 18 | TX-7 | NSD | NSD | NSD | 575000.00 | 5750.00 | 11.50 | |
| 19 | TX-8 | NSD | NSD | NSD | NSD | NSD | NSD | |
| 20 | SJ 1 | NGD | NGD | NGD | 90000.00 | 9000.00 | 19.00 | |
| 21 | SJ-2 | 000.00 | | 50,000.00 | NGD | NGD | NGD | |
| 22 | SJ-3 | 4,000.00 | | 106,000.00 | 575000.00 | 57500.00 | 17.00 | |
| 23 | SJ-4 | NSD | NSD | NSD | NSD | NSD | NSD | |
| 24 | SJ 5 | 150,000.00 | | 1,100,000,000.00 | 57500000.00 | 1925000.00 | 26.00 | |
| 25 | SJ-6 | NSD | NSD | NSD | NSD | NSD | NSD | |
| 26 | SJ-7 | NSD | NSD | NSD | 575000.00 | 3750.00 | 21.00 | |
| 27 | SJ-8 | NSD | NSD | NSD | NSD | NSD | NSD | |
| 28 | TECA 1 | NSD | NGD | NGD | 900000.00 | 67500.00 | 7.00 | |
| 29 | TECA-2 | NSD | NSD | NSD | NSD | NSD | NSD | |
| 30 | TECA-3 | NSD | NSD | NSD | 2275000.00 | 37500.00 | 10.50 | |
| 31 | TECA-4 | 4,900,000.00 | | 5,000,000.00 | NSD | NSD | NSD | |
| 32 | TECA-5 | 23,000,000.00 | | 5,000,000.00 | 2275000.00 | 00000.00 | 10.00 | |
| 33 | TECA-6 | 23,000.00 | | 8,000,000.00 | NSD | NSD | NSD | |
| 34 | TECA-7 | NSD | NSD | NSD | [illegible] | [illegible] | [illegible] | |
| 35 | TECA-8 | NSD | NSD | NSD | NSD | NSD | NSD | |
| 36 | GC-1A | 15,000.00 | | 000,000.00 | 57500.00 | 3500.00 | 5.50 | |
| 37 | GC-1B | NSD | NSD | NSD | NSD | NSD | NSD | |
| 38 | GC-2A | NSD | NSD | NSD | NSD | NSD | NSD | |
| 39 | GC 2B | 2,000.00 | | 000,000.00 | 10750000.00 | 1650000.00 | 6.50 | |
| 40 | GC-3A | NGD | NGD | NGD | 2325000.00 | 3250.00 | 5.00 | |
| 41 | GC-3B | ND | | 450,000.00 | NSD | NSD | NSD | |
| 42 | [illegible] | NSD | NSD | NSD | NSD | NSD | NSD | |

**Conjunto original de datos**

**Diagramas ORM**

Permite tener un análisis de las tablas que conformarán la base de datos. Permitiendo analizar el funcionamiento lógico y almacenamiento de los datos que se desean alojar, administrar y operar.

**Diagrama ORM de la Tabla BTEX**
Número de Valores: 5
Numero de Entidades relacionadas: 1
Claves Externas: 1

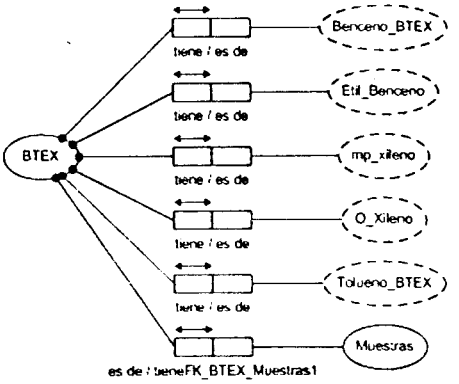



**Diagrama ORM de la Tabla CLASES**
Número de Valores: 3
Número de Entidades relacionadas: 1
Claves Externa: 0

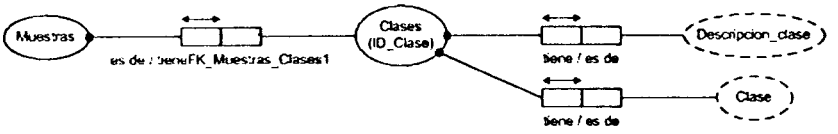
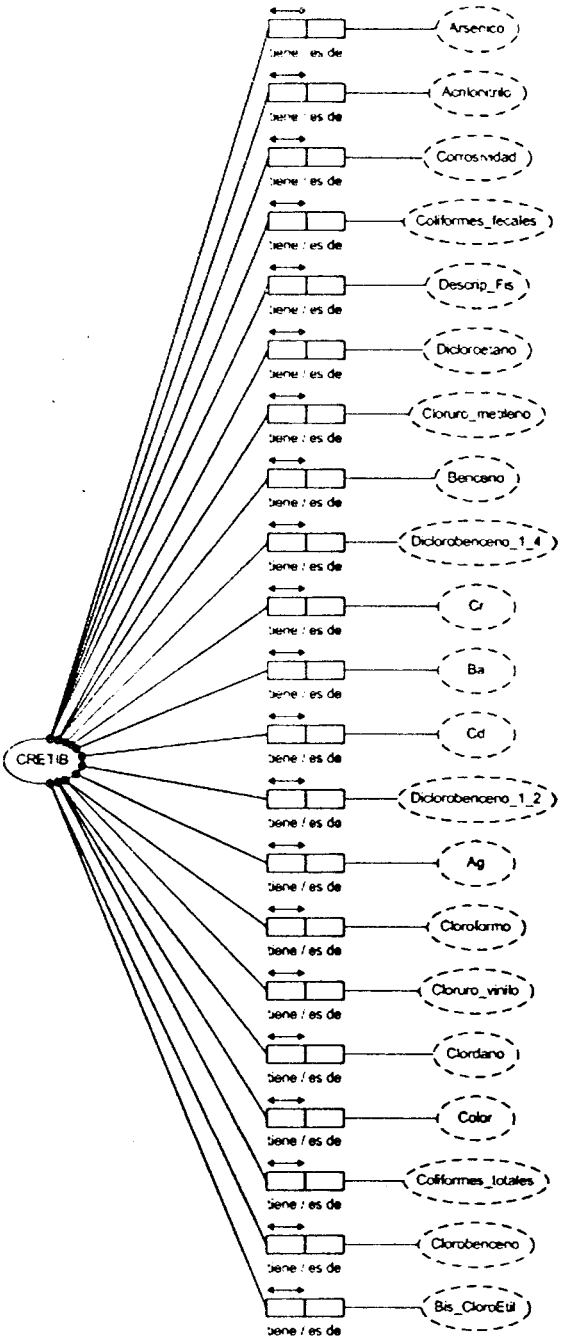



**Diagrama ORM de la Tabla CRETIB**
Número de Valores: 61
Número de Entidades relacionadas: 1
Caves Externas: 1


CRETIB
Arsenico
Acrilonitrilo
Corrosividad
Coliformes_fecales
Descrip_Fis
Dicloroetano
Cloruro_metileno
Benceno
Diclorobenceno_1_4
Cr
Ba
Cd
Diclorobenceno_1_2
Ag
Cloroformo
Cloruro_vinilo
Clordano
Color
Coliformes_totales
Clorobenceno
Bis_CloroEtil

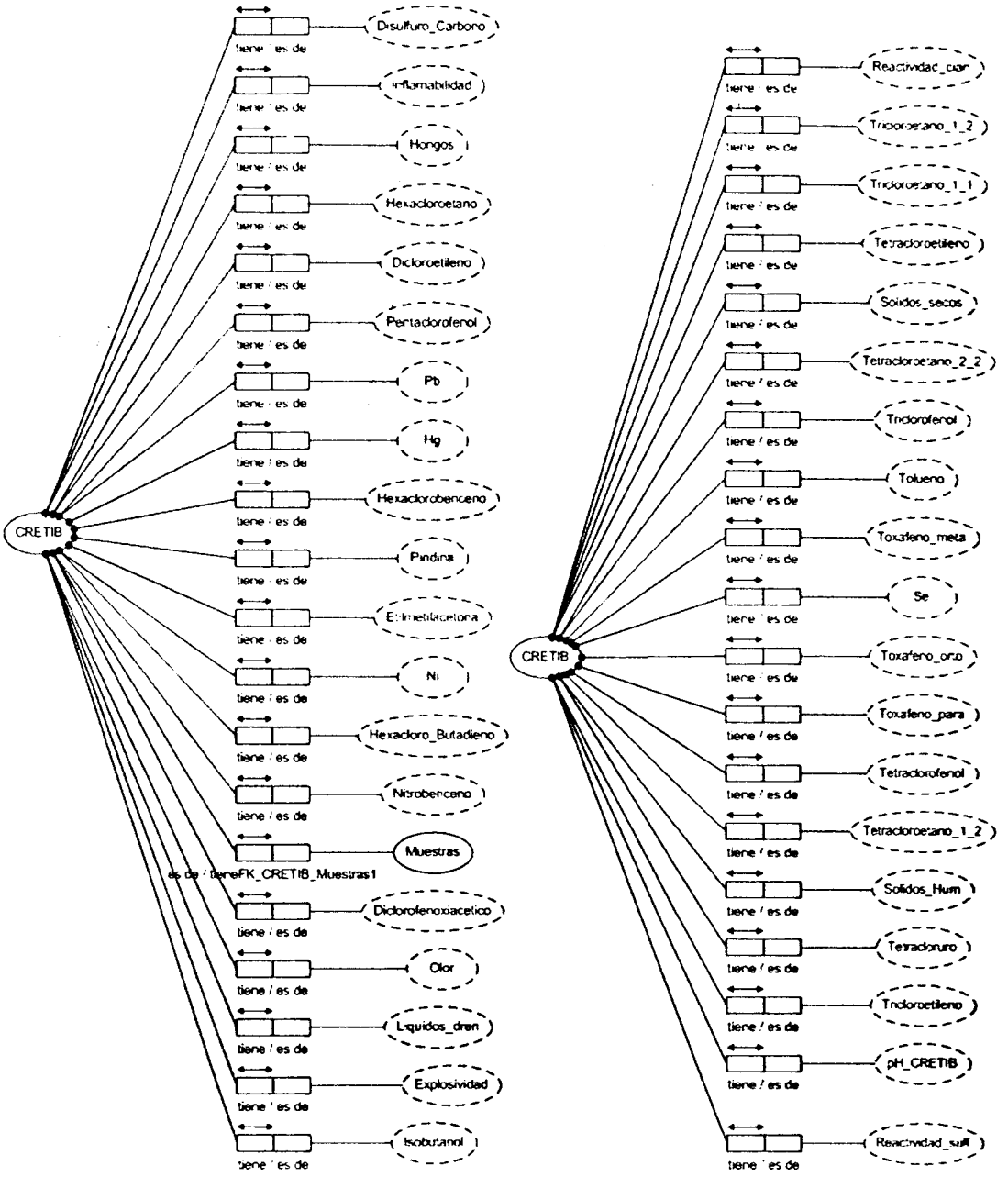

CRETIB
Disulfuro_Carboro
inflamabilidad
Hongos
Hexacloroetano
Dicloroetileno
Pentaclorofenol
Pb
Hg
Hexaclorobenceno
Piridina
Ni
Hexacloro_Butadieno
Nitrobenceno
Muestras
Diclorofenoxiacetico
Olor
Explosividad
Isobutanol
Reactividad_cian
Tricloroetano_1_2
Tricloroetano_1_1
Tetracloroetileno
Solidos_secos
Tetracloroetano_2_2
Triclorofenol
Tolueno
Toxafeno_meta
Se
Toxafeno_orto
Toxafeno_para
Tetraclorofenol
Tetracloroetano_1_2
Solidos_Hum
Tetracloruro
Tricloroetileno
pH_CRETIB
Reactividad_sulf
tiene / es de

**Diagrama ORM de la Tabla METALES_PESADOS**
Número de Valores: 13
Numero de Entidades relacionadas: 1
Claves Externas: 1

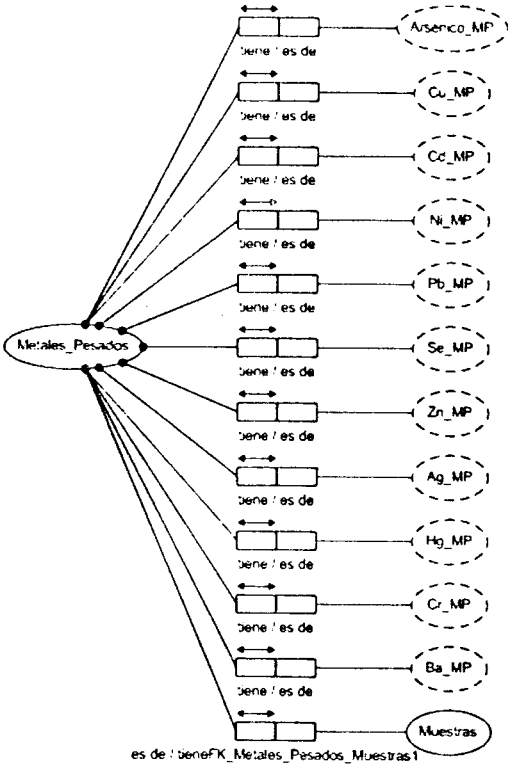


**Diagrama ORM de la Tabla MUESTRAS**
Número de Valores: 13
Numero de Entidades relacionadas: 10
Claves Externar: 2

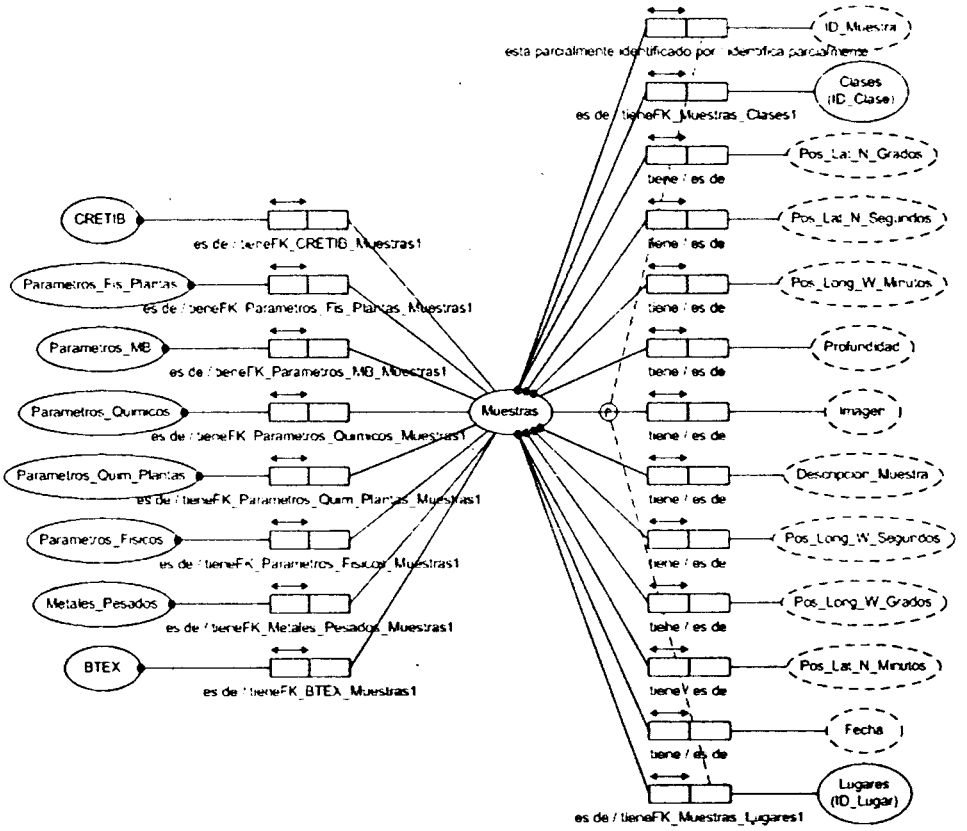


**Diagrama ORM de la Tabla PARAMETROS_FISICOS**
Número de Valores: 9
Numero de Entidades relacionadas: 1
Claves Externar: 1

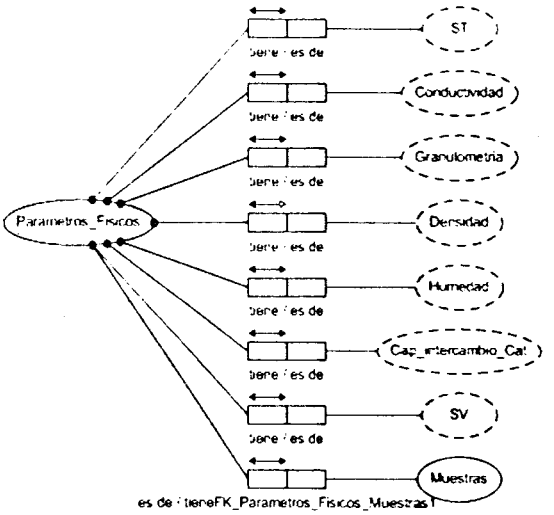



**Diagrama ORM de la Tabla PARAMETROS_FIS_PLANTA**
Número de Valores: 6
Numero de Entidades relacionadas: 1
Claves Externar: 1

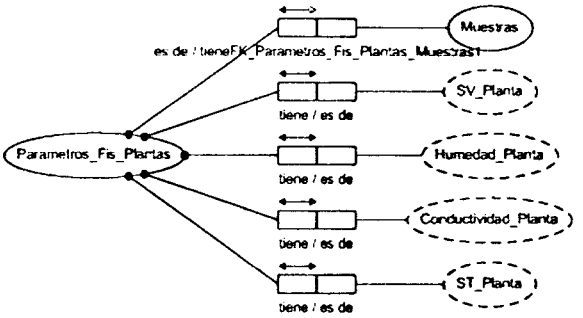


**Diagrama ORM de la Tabla PARAMETROS_QUIMICOS**
Número de Valores: 10
Numero de Entidades relacionadas: 1
Claves Externar: 1

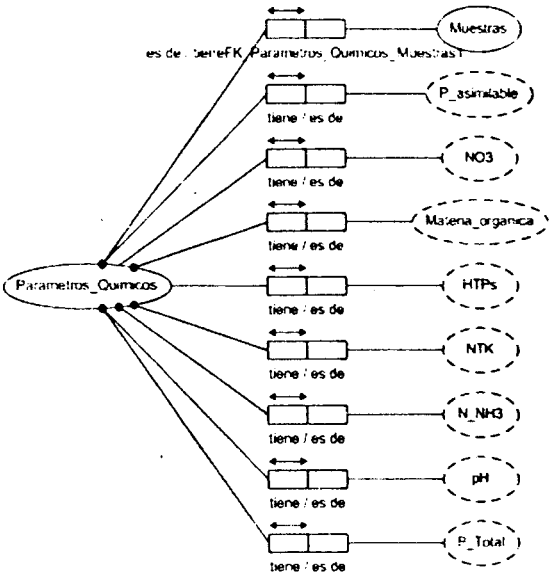



**Diagrama ORM de la Tabla PARAMETROS_QUIM_PLANTAS**
Número de Valores: 8
Numero de Entidades relacionadas: 1
Claves Externar: 1

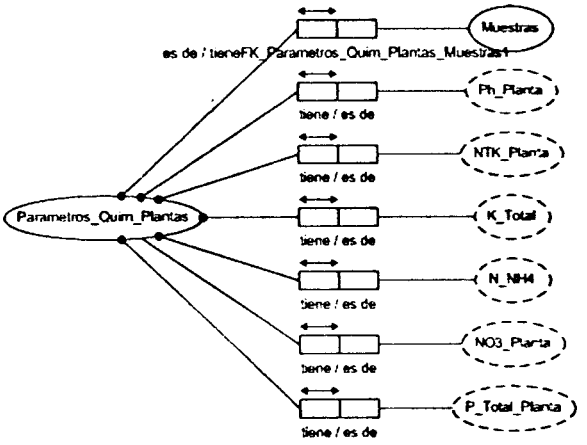


**Diagrama ORM de Tabla de Lugares**
Número de Valores: 4
Numero de Entidades relacionadas: 1
Claves Externar: 0

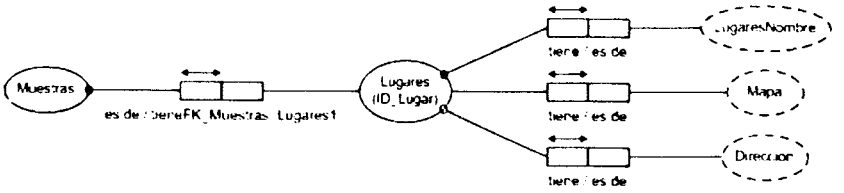
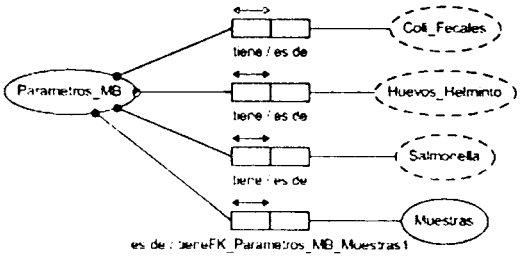



**Diagrama ORM de Tabla de Parametros_MB**
Número de Valores: 5
Numero de Entidades relacionadas: 1
Claves Externar: 1




**Resumen de base de datos**

Este contenido es un apoyo para el diccionario de datos que contempla las características de cada una de las tablas que se emplearán en la construcción de la base de datos, así como las características de nombre, tipo de dato, relación y contenido.

**BTEX**

Propietario: **dbo**
Nombre de BD de destino: Tesis
Número de columnas: **7**
Número de índices: **0**
Número de claves externas: **1**
Códigos: **0**

**Atributos extendidos:**

OnFileGroup PRIMARY
Clustered PK **Sí**

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | int | No está permitida | |
| ID_Lugar | char(10) | No está permitida | |
| Benceno_BTEX | char(10) | No está permitida | ug/kg |
| Tolueno_BTEX | char(10) | No está permitida | mg/kg |
| Etil_Benceno | char(10) | No está permitida | ug/kg |
| O_Xileno | char(10) | No está permitida | ug/kg |
| mp_xileno | char(10) | No está permitida | ug/kg |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_BTEX_Muestras1 | ID_Muestra<br>ID_Lugar | Muestras.ID_Muestra<br>Muestras.ID_Lugar |

**Clases**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 3
Número de índices: 0
Número de claves externas: 0
Códigos: 0

**Atributos extendidos:**

OnFileGroup PRIMARY
TextImageOnGroup PRIMARY
Clustered PK Si

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Clase | int identity | No está permitida | |
| Clase | varchar(50) | No está permitida | |
| Descripcion_clase | Text | No está permitida | Texto |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Muestras_Clases1 | Muestras.ID_Clase | ID_Clase |

**CRETIB**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 61
Número de índices: 0
Número de claves externas: 1
Códigos: 0

**Atributos extendidos:**

OnFileGroup PRIMARY
TextImageOnGroup PRIMARY
Clustered PK Si

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | Int | No está permitida | |
| ID_Lugar | char(10) | No está permitida | |
| Descrip_Fis | char(10) | No está permitida | Texto |
| Color | char(10) | No está permitida | Texto |
| Olor | char(10) | No está permitida | Texto |
| Solidos_Hum | char(10) | No está permitida | % |
| Solidos_secos | char(10) | No está permitida | % |
| Liquidos_dren | char(10) | No está permitida | % |
| pH_CRETIB | numeric(13) | Permitido | |
| Corrosividad | Text | No está permitida | mm/año |
| Inflamabilidad | Text | No está permitida | Texto |
| Reactividad_cian | char(10) | No está permitida | mg/kg |
| Reactividad_sulf | char(10) | No está permitida | mg/kg |
| Explosividad | Text | No está permitida | Texto |
| Arsenico | char(10) | No está permitida | mg/l |
| Ba | char(10) | No está permitida | mg/l |
| Cd | char(10) | No está permitida | mg/l |
| Cr | char(10) | No está permitida | mg/l |
| Ni | char(10) | No está permitida | mg/l |
| Hg | char(10) | No está permitida | mg/l |
| Ag | char(10) | No está permitida | mg/l |
| Pb | char(10) | No está permitida | mg/l |
| Se | char(10) | No está permitida | mg/l |
| Benceno | char(10) | No está permitida | mg/l |
| Tolueno | char(10) | No está permitida | mg/l |
| Cloroformo | char(10) | No está permitida | mg/l |
| Cloruro_metileno | char(10) | No está permitida | mg/l |
| Cloruro_vinilo | char(10) | No está permitida | mg/l |
| Dicloroetano | char(10) | No está permitida | mg/l |
| Dicloroetileno | char(10) | No está permitida | mg/l |
| Tetracloroetano_1_2 | char(10) | No está permitida | mg/l |
| Tetracloroetano_2_2 | char(10) | No está permitida | mg/l |
| Tetracloruro | char(10) | No está permitida | mg/l |
| Tetracloroetileno | char(10) | No está permitida | mg/l |
| Tricloroetileno | char(10) | No está permitida | mg/l |
| Tricloroetano_1_1 | char(10) | No está permitida | mg/l |
| Tricloroetano_1_2 | char(10) | No está permitida | mg/l |
| Clorobenceno | char(10) | No está permitida | mg/l |
| Acrilonitrilo | char(10) | No está permitida | mg/l |
| Bis_CloroEtil | char(10) | No está permitida | mg/l |
| Disulfuro_Carbono | char(10) | No está permitida | mg/l |
| Isobutanol | char(10) | No está permitida | mg/l |
| Piridina | char(10) | No está permitida | mg/l |
| Etilmetilacetona | char(10) | No está permitida | mg/l |
| Hexaclorobenceno | char(10) | No está permitida | mg/l |
| Diclorobenceno_1_2 | char(10) | No está permitida | mg/l |
| Diclorobenceno_1_4 | char(10) | No está permitida | mg/l |
| Hexacloro_Butadieno | char(10) | No está permitida | mg/l |
| Nitrobenceno | char(10) | No está permitida | mg/l |
| Toxafeno_orto | char(10) | No está permitida | mg/l |
| Toxafeno_meta | char(10) | No está permitida | mg/l |
| Toxafeno_para | char(10) | No está permitida | mg/l |
| Pentaclorofenol | char(10) | No está permitida | mg/l |
| Tetraclorofenol | char(10) | No está permitida | mg/l |
| Triclorofenol | char(10) | No está permitida | mg/l |
| Clordano | char(10) | No está permitida | mg/l |
| Diclorofenoxiacetico | char(10) | No está permitida | mg/l |

| | | | |
|---|---|---|---|
| Hexacloroetano | char(10) | No está permitida | mg/l |
| Coliformes_fecales | char(10) | No está permitida | NMP/g B.H. |
| Coliformes_totales | char(10) | No está permitida | NMP/g B.H. |
| Hongos | char(10) | No está permitida | Col/g B.H. |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_CRETIB_Muestras1 | ID_Muestra<br>ID_Lugar | Muestras.ID_Muestra<br>Muestras.ID_Lugar |

**Lugares**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 5
Número de índices: 0
Número de claves externas: 1
Códigos: 0

**Atributos extendidos:**

OnFileGroup PRIMARY
TextImageOnGroup PRIMARY
Clustered PK Sí

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Lugar | char(10) | No está permitida | |
| ID_Clase | Int(4) | No está permitida | |
| Nombre | Text | No está permitida | Texto |
| Direccion | Text | No está permitida | Texto |
| Mapa | Text | Permitido | Texto |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Muestras_Lugares1 | Muestras.ID_Lugar<br>Clases.ID_Clase | ID_Lugar |

**Metales_Pesados**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 13
Número de índices: 0
Número de claves externas: 1
Códigos: 0

**Atributos extendidos:**

OnFileGroup PRIMARY
Clustered PK Sí

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | Int | No está permitida | |

| | | | |
|---|---|---|---|
| ID_Lugar | char(10) | No está permitida | |
| Arsenico_MP | char(10) | No está permitida | mg/kg |
| Ba_MP | char(10) | No está permitida | mg/kg |
| Cu_MP | char(10) | No está permitida | mg/kg |
| Cr_MP | char(10) | No está permitida | mg/kg |
| Cd_MP | char(10) | No está permitida | mg/kg |
| Hg_MP | Varchar(10) | Permitido | mg/kg |
| Ni_MP | char(10) | No está permitida | mg/kg |
| Ag_MP | char(10) | No está permitida | mg/kg |
| Pb_MP | char(10) | No está permitida | mg/kg |
| Zn_MP | char(10) | No está permitida | mg/kg |
| Se_MP | char(10) | No está permitida | mg/kg |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Metales_Pesados_Muestras1 | ID_Muestra<br>ID_Lugar | Muestras.ID_Muestra<br>Muestras.ID_Lugar |

**Muestras**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 12
Número de índices: 0
Número de claves externas: 1
Códigos: 0

**Atributos extendidos:**

OnFileGroup PRIMARY
TextImageOnGroup PRIMARY
Clustered PK Sí

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | Int | No está permitida | |
| ID_Lugar | char(10) | No está permitida | |
| Fecha | Datetime | No está permitida | Fecha |
| Pos_Lat_N_Grados | Numeric(13) | No está permitida | Adimensional |
| Pos_Lat_N_Minutos | Numeric(13) | No está permitida | Adimensional |
| Pos_Lat_N_Segundos | Numeric(13) | No está permitida | Adimensional |
| Pos_Long_W_Grados | Numeric(13) | No está permitida | Adimensional |
| Pos_Long_W_Minutos | Numeric(13) | No está permitida | Adimensional |
| Pos_Long_W_Segundos | Numeric(13) | No está permitida | Adimensional |
| Profundidad | Numeric(13) | Permitido | M |
| Descripcion_Muestra | Text | No está permitida | Texto |
| Imagen | Text | Permitido | Imagen |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Muestras_Lugares1 | ID_Lugar | Lugares.ID_Lugar |
| FK_BTEX_Muestras1 | BTEX.ID_Muestra<br>BTEX.ID_Lugar | ID_Muestra<br>ID_Lugar |
| FK_CRETIB_Muestras1 | CRETIB.ID_Muestra<br>CRETIB.ID_Lugar | ID_Muestra<br>ID_Lugar |
| FK_Metales_Pesados_Muestras1 | Metales_Pesados.ID_Muestra<br>Metales_Pesados.ID_Lugar | ID_Muestra<br>ID_Lugar |
| FK_Parametros_Fis_Plantas_Muestras1 | Parametros_Fis_Plantas.ID_Muestra<br>Parametros_Fis_Plantas.ID_Lugar | ID_Muestra<br>ID_Lugar |
| FK_Parametros_Fisicos_Muestras1 | Parametros_Fisicos.ID_Muestra<br>Parametros_Fisicos.ID_Lugar | ID_Muestra<br>ID_Lugar |
| FK_Parametros_MB_Muestras1 | Parametros_MB.ID_Muestra<br>Parametros_MB.ID_Lugar | ID_Muestra<br>ID_Lugar |
| FK_Parametros_Quim_Plantas_Muestras1 | Parametros_Quim_Plantas.ID_Muestra<br>Parametros_Quim_Plantas.ID_Lugar | ID_Muestra<br>ID_Lugar |
| FK_Parametros_Quimicos_Muestras1 | Parametros_Quimicos.ID_Muestra<br>Parametros_Quimicos.ID_Lugar | ID_Muestra<br>ID_Lugar |

**Parametros_Fis_Plantas**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 6
Número de índices: 0
Número de claves externas: 1
Códigos: 0

**Atributos extendidos:**

OnFileGroup PRIMARY
Clustered PK Sí

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | int | No está permitida | |
| ID_Lugar | char(10) | No está permitida | |
| ST_Planta | Numeric(13) | No está permitida | % |
| SV_Planta | Numeric(13) | No está permitida | % |
| Conductividad_Planta | Numeric(13) | No está permitida | mS/cm |
| Humedad_Planta | Numeric(13) | No está permitida | % |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Parametros_Fis_Plantas_Muestras1 | ID_Muestra<br>ID_Lugar | Muestras.ID_Muestra<br>Muestras.ID_Lugar |

**Parametros_Fisicos**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 9
Número de índices: 0
Número de claves externas: 1
Códigos: 0

**Atributos extendidos:**

OnFileGroup PRIMARY
TextImageOnGroup PRIMARY
Clustered PK Sí

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | int | No está permitida | |
| ID_Lugar | char(10) | No está permitida | |
| ST | Numeric(13) | No está permitida | % |
| SV | Numeric(13) | No está permitida | % |
| Conductividad | Numeric(13) | No está permitida | mS/cm |
| Cap_intercambio_Cat | Numeric(13) | No está permitida | meq/100g |
| Granulometria | Text | No está permitida | Textura |
| Humedad | Numeric(13) | No está permitida | % |
| Densidad | Numeric(13) | No está permitida | gr/cm3 |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Parametros_Fisicos_Muestras1 | ID_Muestra<br>ID_Lugar | Muestras.ID_Muestra<br>Muestras.ID_Lugar |

**Parametros_MB**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 5
Número de índices: 0
Número de claves externas: 1
Códigos: 0

**Atributos extendidos:**

OnFileGroup PRIMARY
Clustered PK Sí

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | int | No está permitida | |
| ID_Lugar | char(10) | No está permitida | |
| Coli_Fecales | char(10) | No está permitida | NMP/gr ST(B.S.) |
| Salmonella | char(10) | No está permitida | NMP/gr ST(B.S.) |
| Huevos_Helminto | char(10) | No está permitida | NMP/gr ST(B.S.) |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Parametros_MB_Muestras1 | ID_Muestra<br>ID_Lugar | Muestras.ID_Muestra<br>Muestras.ID_Lugar |

**Parametros_Quim_Plantas**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 8
Número de índices: 0
Número de claves externas: 1
Códigos: 0
**Atributos extendidos:**

OnFileGroup PRIMARY
Clustered PK Sí

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | int | No está permitida | |
| ID_Lugar | char(10) | No está permitida | |
| P_Total_Planta | char(10) | No está permitida | mg/kg |
| Ph_Planta | numeric(13) | No está permitida | mg/kg |
| NO3_Planta | varchar(10) | Permitido | mg/kg |
| NTK_Planta | numeric(13) | No está permitida | mg/kg |
| N_NH4 | char(10) | No está permitida | mg/kg |
| K_Total | char(10) | No está permitida | mg/kg |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Parametros_Quim_Plantas_Muestras1 | ID_Muestra<br>ID_Lugar | Muestras.ID_Muestra<br>Muestras.ID_Lugar |

**Parametros_Quimicos**

Propietario: dbo
Nombre de BD de destino: Tesis
Número de columnas: 10
Número de índices: 0
Número de claves externas: 1
Códigos: 0

**Atributos extendidos:**
OnFileGroup PRIMARY
Clustered PK Sí

| Columnas | Tipo de datos | Permitir valores NULL | Unidad |
|---|---|---|---|
| ID_Muestra | Int | No está permitida | |
| ID_Lugar | char(10) | No está permitida | |
| P_Total | char(10) | No está permitida | mg/kg |
| P_asimilable | numeric(13) | No está permitida | mg/kg |
| pH | numeric(13) | No está permitida | |
| NO3 | varchar(10) | Permitido | mg/kg |
| N_NH3 | char(10) | No está permitida | mg/kg |
| Materia_organica | numeric(13) | No está permitida | % |
| NTK | numeric(13) | No está permitida | mg/kg |
| HTPs | char(10) | Permitido | |

| Claves externas | Secundaria | Primario |
|---|---|---|
| FK_Parametros_Quimicos_Muestras1 | ID_Muestra<br>ID_Lugar | Muestras.ID_Muestra<br>Muestras.ID_Lugar |

# GLOSARIO

**ACL.** Siglas de Access Control Lists: Listas de control de acceso. Se deben establecer las listas ACL adecuadas en los recursos apropiados para permitir que únicamente los clientes indicados obtengan acceso.

**ACTIVE DIRECTORY.** Active Directory es el servicio de directorio LDAP que utiliza el sistema operativo Windows 2000.

**ALGORITMO.** Conjunto de reglas claramente definidas para la resolución de una determinada clase de problemas. La escritura de un programa es sencillamente la elaboración de un algoritmo adecuado para la resolución del problema planteado. Un programa de software es la transcripción, en lenguaje de programación, de un algoritmo.

**ALMACENAR.** Incluir los datos en una memoria, externa o interna a la computadora, adecuada para conservarlos. Sinónimos de este término son escribir, guardar, grabar y salvar.

**ALMACÉN DE CERTIFICADOS.** Un almacén de certificados es una ubicación de almacenamiento de certificados, listas de revocaciones de certificados (CRL) y listas de certificados de confianza (CTL).

**ALMACÉN DE CLAVES.** Un almacén de claves es la ubicación donde la API de criptografía de Microsoft (CryptoAPI) almacena los pares de claves (normalmente en un archivo o en una clave del Registro). Los almacenes de claves son específicos de un usuario o del equipo en el que se generaron las claves.

**ANSI.** Siglas de American National Standard Institute (instituto nacional americano de estándares). Se trata de una organización norteamericana que se encarga de la formulación de normas en diversos sectores técnicos. En Windows es el juego de códigos empleado para definir los caracteres que se introducen en los documentos.

**ANTIVIRUS.** Programa que busca y eventualmente elimina los virus informáticos que pueden haber infectado un disco rígido o disquete.

**APLICACIÓN.** Es el problema o conjunto de problemas para los que se diseña una solución mediante computadora. Ejemplos de aplicaciones son los procesadores de texto (procesamiento o tratamiento de la palabra), las bases de datos (organización y procesamiento de datos) y las hojas de cálculo (organización y procesamiento de números). En Windows se emplea este término indistintamente con el de programa.

**ARCHIVO.** Es un conjunto de datos relacionados de manera lógica, como puede ser el conjunto de los nombres, direcciones y teléfonos de los empleados de una empresa determinada.

**ARRANCAR.** Poner en marcha una computadora o un programa.

**ATAQUE DE DICCIONARIO.** Un ataque de diccionario es un ataque por fuerza bruta en el que el atacante prueba todas las claves secretas posibles para descifrar datos cifrados. Puede reducir la amenaza de esta forma de ataque mediante el uso de un valor salt junto con datos cifrados (o hash).

**AT&T.** American Telephone and Telegraph Corporation. (Corporación Americana de Telefonía y Telegrafía.)

**AUI.** Asociación de usuarios de Internet.

**AUTENTICACIÓN.** La autenticación es el proceso de comprobación de la identidad. Por ejemplo, cuando se inicia una sesión en Windows, el sistema operativo autentica al usuario mediante la solicitud de sus credenciales: un nombre de usuario y una contraseña.

**AUTENTICACIÓN ANÓNIMA.** La autenticación anónima es una forma de autenticación de IIS en la que IIS no intenta comprobar la identidad de sus clientes. La autenticación anónima es similar a la no autenticación. A menudo se utiliza junto con la autenticación mediante Formularios de ASP.NET, que utiliza un formulario HTML para capturar las credenciales del cliente.

**AUTENTICACIÓN BÁSICA.** La autenticación básica forma parte del protocolo HTTP 1.0. Su uso es muy extendido, porque la implementan prácticamente todos los servidores Web y exploradores Web. La autenticación básica es un mecanismo de autenticación sencillo que no requiere criptografía ni protocolos de enlace desafío/respuesta. En lugar de eso, las credenciales de un principal (nombre de usuario y contraseña) se pasan directamente del cliente al servidor. La autenticación básica no es segura, a menos que se combine con SSL, ya que la contraseña no se cifra antes de transmitirla a la red. Se transmite mediante codificación base 64, de forma que puede obtenerse fácilmente la contraseña de texto sin cifrar.

**AUTENTICACIÓN DE CERTIFICADOS.** La autenticación de certificados es una forma de autenticación de IIS en la que IIS acepta certificados de cliente que se utilizan para comprobar la identidad del cliente. Con esta forma de autenticación, IIS puede optar por asignar un certificado de cliente a una cuenta de usuario de Windows mediante una tabla de asignación interna o Active Directory.

**AUTENTICACIÓN IMPLÍCITA.** La autenticación implícita viene definida por el protocolo HTTP 1.1, aunque su uso no es muy extendido. Con esta forma de autenticación, una contraseña de texto sin cifrar no se transmite a través de la red. En su lugar, se transmite una contraseña hash o implícita. Aunque es más segura que la autenticación básica, requiere tener instalado Internet Explorer 5.0 o posterior en el

cliente, y un equipo con Windows 2000 ejecutando IIS 5.0 con Active Directory en el servidor.

**AUTENTICACIÓN MEDIANTE FORMULARIOS.** La autenticación mediante formularios es un tipo autenticación admitida por ASP.NET que requiere que los usuarios inicien la sesión proporcionando credenciales de inicio de sesión a través de un formulario HTML.

**AUTENTICACIÓN MUTUA.** La autenticación mutua es una forma de autenticación en la que el cliente autentica al servidor, a la vez que el servidor autentica al cliente. La autenticación mutua no es admitida por NTLM pero sí por Kerberos. También es posible con SSL cuando el servidor acepta o solicita certificados de cliente.

**AUTORIDAD.** Una autoridad es una entidad (organización o equipo) de confianza que se utiliza para proporcionar servicios de autenticación.

**AUTORIZACIÓN.** La autorización es el proceso que determina si se permite o no a una identidad autenticada tener acceso a un recurso solicitado o ejecutar una operación solicitada.

**BACKUP.** Copia de seguridad. Se hace para prevenir una posible pérdida de información.

**BASE DE DATOS SAM.** La base de datos SAM es la base de datos que utilizan Windows NT y Windows 2000 (sin Active Directory) para mantener las cuentas de usuario y de grupo

**BINARIO.** Es un sistema de numeración en el que los dígitos se representan utilizando únicamente dos cifras, 0 y 1. Como adjetivo indica dos opciones alternativas.

**BIOS.** Siglas de Basic Input/Output System (sistema básico de entrada/salida). Es un programa cargado en ROM por el fabricante que gestiona la configuración básica del sistema. Entre otras cosas, se emplea para controlar los procesos de entrada y salida entre una computadora y sus periféricos.

**BOOT.** (bootear) cargar el sistema operativo de una computadora.

**BYTE.** Unidad de información utilizada por las computadoras. Cada byte está compuesto por ocho bits.

**CABECERA.** Encabezamiento de un impreso o documento. También se aplica a la información preliminar incluida al comienzo de un bloque de datos relativa al bloque siguiente. En comunicaciones es un bloque de caracteres que indica las características del mensaje.

**CACHÉ.** Un mecanismo especial de almacenamiento de alta velocidad. Puede ser una sección reservada de memoria principal o un dispositivo de almacenamiento de alta velocidad independiente. Dos tipos de cache son utilizados comúnmente en las computadoras personales: cache de memoria y cache de disco.

La memoria cache es una porción de memoria hecha de RAM estática de alta velocidad (SRAM) en vez de la más lenta y barata RAM dinámica (DRAM) usada como memoria principal. Utilizar memoria cache es efectivo porque la mayoría de los programas acceden a los mismos datos o instrucciones una y otra vez. Al mantener la mayor cantidad posible de esta información en la SRAM, la computadora evita acceder a la DRAM que es más lenta.

A veces la memoria cache es construida dentro de la arquitectura de los microprocesadores, a tal memoria se le denomina cache Nivel 1 o Level 1 (L1). La mayoría de las PCs actuales cuentan además con una memoria cache externa llamada cache Nivel 2 o Level 2 (L2) que se encuentra entre el CPU y la DRAM.

La cache de disco trabaja bajo el mismo principio que la cache de memoria, pero en vez de usar SRAM de alta velocidad, utiliza memoria principal convencional. Los datos recientemente accedidos del disco son almacenados en un buffer de memoria. Cuando un programa necesita acceder a datos del disco primero verifica el cache de disco para ver si los datos se encuentran ahí. Esto conlleva una mejora dramática del rendimiento ya que acceder a un byte de datos en RAM puede ser cientos de veces más rápido que acceder un byte en el disco duro.

**CD-ROM.** Siglas de Compact Disc Read Only Memory (memoria de sólo lectura en disco compacto). Es un soporte de almacenamiento masivo de datos basado en los discos compactos de audio, que registran la información en el disco mediante láser. No permite la modificación de los datos registrados.

**CERTIFICADO.** Un certificado es una estructura de datos firmados digitalmente que contiene información acerca de un sujeto (persona o aplicación) y de la clave pública del mismo. Las organizaciones de confianza denominadas entidades emisoras de certificados (CA) emiten los certificados tras verificar la identidad del sujeto.

**CERTIFICADO DE CLIENTE.** Un certificado de cliente es un certificado que utilizan los clientes para proporcionar a las aplicaciones servidor una identificación positiva de su identidad.

**CG.** Computer Graphics. Gráficos de Computador.

**CGI.** Common Gateway Interface. Interfaz de Acceso Común. Programas usados para hacer llamadas a rutinas o controlar otros programas o bases de datos desde una página Web. También pueden generar directamente HTML.

**CIFRADO.** El cifrado es un algoritmo criptográfico que se utiliza para cifrar datos.

**CIFRADO DE CLAVES PÚBLICA Y PRIVADA.** El cifrado de claves pública y privada es una forma asimétrica de cifrado basado en un par de claves, pública y privada, generadas criptográficamente. Los

datos cifrados con una clave privada pueden descifrarse únicamente con la clave pública correspondiente y viceversa.

**CIFRADO SIMÉTRICO.** El cifrado simétrico es una forma de cifrado que utiliza la misma clave (única) para cifrar y descifrar datos. Tanto el emisor como el destinatario de los datos cifrados deben tener la misma clave.

**CIFRAR.** Cifrar es convertir datos (texto sin formato) en un valor aparentemente aleatorio y sin sentido (texto cifrado), que es difícil de descifrar sin una clave secreta. Se utiliza para proporcionar confidencialidad a los mensajes.

**CLASE.** En programación orientada a objetos, una clase es una categoría de objetos. Por ejemplo, podría haber una clase de objetos llamada forma que contenga objetos como círculos, rectángulos, y triángulos. La clase define todas las propiedades comunes de los diferentes objetos que pertenecen a ella. En los lenguajes .NET, las clases son plantillas usadas para definir nuevos tipos. Las clases describen tanto las propiedades como los comportamientos de los objetos. Las propiedades contienen los datos que son expuestos por la clase. Los comportamientos son la funcionalidad del objeto, y son definidos por los métodos públicos (también llamados función miembro) y los eventos de la clase. Colectivamente, las propiedades públicas y los métodos de una clase son conocidos como la interfaz del objeto. Las clases por sí mismas no son objetos, pero son utilizadas para crear objetos en memoria.

**CLAVE.** Una clave es un valor suministrado por un algoritmo de cifrado o descifrado que se utiliza para cifrar y descifrar datos. Los algoritmos de cifrado simétricos utilizan la misma clave para cifrar y descifrar, mientras que los algoritmos asimétricos utilizan un par de claves: pública y privada.

**CLAVE DE SESIÓN.** La clave de sesión es una clave simétrica generada aleatoriamente que se utiliza para cifrar datos que se transmiten entre dos partes. Las claves de sesión se utilizan una sola vez (en una sola sesión) y, a continuación, se descartan.

**CLAVE PRIVADA.** Una clave privada es la mitad del secreto de un par de claves que se utiliza en un algoritmo de clave pública. Las claves privadas suelen utilizarse para cifrar una clave de sesión simétrica, para firmar digitalmente un mensaje o para descifrar un mensaje que haya sido cifrado con la clave pública correspondiente.

**CLAVE PÚBLICA.** Una clave pública es la mitad pública de un par de claves: pública y privada. Suele utilizarse al descifrar una clave de sesión o una firma digital. La clave pública también se puede utilizar para cifrar un mensaje, garantizando de este modo que sólo pueda descifrar el mensaje la persona con la correspondiente clave privada.

**CODIFICACIÓN BASE 64.** La codificación base 64 es un método bien definido para procesar datos binarios como texto ASCII imprimible, que se puede utilizar adecuadamente con los protocolos basados en texto como HTTP. No es un tipo de cifrado.

**CÓDIGO.** Es un conjunto de símbolos y reglas que sirven para representar datos de forma que puedan ser reconocidos por una computadora.

**CÓDIGO HAMMING.** Es un sistema de detección y corrección automática de errores en información electrónica. De lo que se trata, explicado básicamente, es de asociar una serie de bits de validación o paridad a los bits de datos, de tal forma que una alteración en cualquiera de esos bits de datos pueda ser detectada y corregida adecuadamente.

**COMANDO.** Término que define una instrucción, mandato u orden dado a la computadora mediante el cual el usuario le informa de las operaciones o tareas que quiere realizar con su ayuda.

**COMUNICACIÓN SEGURA.** La comunicación segura consiste en proporcionar integridad y privacidad a los mensajes cuando se transmiten datos a través de una red. Algunas tecnologías que proporcionan una comunicación segura son SSL e IPSec.

**CONFIANZA.** Los sistemas seguros se basan en el concepto de confianza en mayor o menor medida. Por ejemplo, se confía en que los usuarios con privilegios administrativos (es decir, los administradores) administrarán correctamente un sistema y no ejecutarán acciones malintencionadas deliberadamente. De forma similar, se debe confiar en el código que se ejecuta con privilegios extendidos, como los controladores de dispositivos, y en el código que se ejecuta como LocalSystem. El código que requiere tal confianza de manera implícita se ejecuta en la base de computación de confianza (TCB) del equipo. No se debe permitir la ejecución del código que no sea de total confianza en la TCB.

**CONFIANZA TRANSITIVA.** La confianza transitiva es una forma bidireccional de relación de confianza entre equipos y dominios. Transitivo significa que si la autoridad A confía en la autoridad B y la autoridad B confía en la autoridad C, entonces la autoridad A confía de manera implícita en la autoridad C (sin que tenga que existir una relación de confianza explícita entre A y C). Active Directory en Windows 2000 admite las relaciones de confianza transitiva.

**CONFIGURACIÓN.** Es un conjunto de opciones que se seleccionan antes de empezar a trabajar con un dispositivo y que sirven para especificar precisamente su modo de funcionamiento, adaptándolo a las especiales condiciones de su hardware.

**CONFIGURAR.** Desde el punto de vista de software, se refiere a establecer, desde un programa

especial, las características de un dispositivo periférico; desde el punto de vista de hardware, consiste en personalizar físicamente dichas características.

**CPU.** Siglas de Central Processing Unit (unidad central de proceso). También llamado procesador, es el núcleo y componente principal de una computadora y permite controlar y procesar todas las operaciones realizadas. Parte de la computadora que contiene el procesador central. También se aplica este término al mismo procesador.

**CRACKER.** Individuo con amplios conocimientos informáticos que desprotege/piratea programas o produce daños en sistemas o redes.

**CRC.** Son las siglas de Cyclic Redundancy Control (control de redundancia cíclica). Es un método de comprobar si una transmisión de datos se ha producido o no correctamente.

**CREDENCIALES.** Las credenciales son el conjunto de elementos que utiliza un objeto principal para probar su identidad. Un ejemplo habitual de conjunto de credenciales es el nombre de usuario y la contraseña.

**CRIPTOGRAFÍA.** La criptografía es el arte y la ciencia de la seguridad de la información. Abarca la confidencialidad, la integridad y la autenticación.

**CUENTA DE SERVICIO.** Una cuenta de servicio es una cuenta configurada específicamente (también conocida como cuenta de proxy), que se utiliza exclusivamente para tener acceso a un recurso indirecto (a menudo una base de datos) en una aplicación distribuida de varios niveles. Los componentes de nivel medio suelen utilizar un número limitado de cuentas de servicio para conectarse a una base de datos y proporcionar la agrupación de la conexión. Las cuentas de servicio pueden ser cuentas de Windows mantenidas en Active Directory o la base de datos SAM, o cuentas SQL mantenidas en SQL Server.

**CUENTA LOCAL.** Una cuenta local es una cuenta de Windows que se mantiene y almacena en la base de datos SAM local de un equipo determinado. A diferencia de las cuentas de dominio, las cuentas locales no se pueden utilizar para tener acceso a los recursos de la red, a menos que se cree una cuenta local duplicada (con el mismo nombre y la misma contraseña) en el equipo remoto.

**CUENTAS DE DOMINIO.** Las cuentas de dominio son cuentas de Windows o de grupo que se mantienen y administran de forma centralizada en una base de datos SAM de un controlador de dominio o en Active Directory.

**DATO.** Es un término genérico empleado para designar números, letras u otros caracteres existentes en una computadora o en su memoria y sobre los cuales actúan los programas.

**DATOS.** Cualquier información que pueda ser usada para cálculo, comparación u otro procesamiento o que requiera ser recordada para un uso futuro. Algunas veces, se usa para referirse a registros u otra información involucrada en un programa a diferencia del programa mismo.

**DBA.** Siglas de Database Administrador, Administrador de la base de datos: Persona que diseña y mantiene la base de datos.

**DBMS.** Siglas de Database Management System, - Sistema Manejador de bases de datos.

**DDL.** Siglas de Data Definition Language. Un lenguaje de definición de datos.

**DERECHO DE ACCESO.** Un derecho de acceso es un atributo que determina el tipo de operación que un determinado grupo o usuario de Windows puede realizar en un objeto seguro. Algunos ejemplos de derechos de acceso son: lectura, escritura, eliminación, ejecución, etc.

**DES.** (Estándar de cifrado de datos). DES es un cifrado de bloques que cifra datos en bloques de 64 bits. DES es un algoritmo simétrico que utiliza el mismo algoritmo y la misma clave para cifrar y descifrar. DES ha sido reemplazado por DES triple.

**DES TRIPLE.** Se trata del cifrado DES triple (3DES). Es una variación del algoritmo de cifrado de bloques DES que cifra el texto sin formato con una clave, cifra el texto cifrado resultante con una segunda clave y, por último, cifra el resultado del segundo cifrado con una tercera clave. DES triple es un algoritmo simétrico que utiliza el mismo algoritmo y las mismas claves para cifrar y descifrar.

**DESCRIPTOR DE SEGURIDAD (SD).** Un descriptor de seguridad (SD) contiene información de seguridad asociada a un objeto asegurable como un archivo o proceso. Un descriptor de seguridad contiene atributos que incluyen una identificación del propietario del objeto, los grupos de seguridad a los que pertenece y dos listas de control de acceso (ACL): la lista de control de acceso discrecional (DACL) que define los derechos de acceso de cada usuario y grupo de usuarios, y la lista de control de acceso al sistema (SACL) que define los tipos de operación que se ejecutan en el objeto y que deberían generar mensajes de auditoría.

**DIRECTORIO RAÍZ.** Es el directorio de nivel superior de un disco que se crea en el momento de dar formato al disco. A partir de este directorio se pueden crear otros directorios y subdirectorios, así como ficheros. También llamado directorio principal.

**DISCO.** Placa recubierta de material magnético que permite almacenar información. Se le llama así por su forma.

**DISCO DURO.** También llamado disco rígido o disco fijo. Es el soporte de almacenamiento de información más utilizado en las computadoras, por

su gran capacidad. Normalmente suele ser interno a la computadora. Es un dispositivo muy delicado formado por una serie de discos apilados uno encima del otro y acomodados en un compartimiento estanco en los que se graban los datos.

**DISCO ÓPTICO.** Es un tipo de soporte de información que no emplea técnicas de grabación magnética. En él la lectura y escritura de datos se realiza mediante rayos de luz láser. Tienen una capacidad de almacenamiento muy superior a la de los discos flexibles, pero son bastante más caros y mucho menos fiables.

**DML.** Siglas de Data Manipulation Language. Un lenguaje de manipulación de datos.

**DOS.** Siglas de Disk Operating System (sistema operativo de disco). Es uno de los tipos de sistema operativo más utilizado en computadoras. Se emplea generalmente para el control de las unidades de disco.

**EFS.** (SISTEMA DE ARCHIVOS DE CIFRADO). El sistema de archivos de cifrado (EFS) se incluye en los sistemas operativos Windows 2000 y posterior para proporcionar la capacidad de cifrado de archivos en un volumen NTFS.

**ENTIDAD EMISORA DE CERTIFICADOS (CA).** Una CA es una organización o entidad de confianza que emite certificados.

**ENTRADA DE CONTROL DE ACCESO (ACE).** Una entrada de control de acceso (ACE) identifica a un usuario o grupo de usuarios concreto en una lista de control de acceso y especifica los derechos de acceso de dicho usuario o grupo de usuarios. Una entrada ACE puede denegar o conceder derechos de forma explícita.

**ENTROPÍA.** La entropía es una medida de incertidumbre. Se utiliza asociada a algunas tecnologías de cifrado para introducir un grado de aleatoriedad en el proceso de cifrado. Un valor de entropía que se utiliza con una clave para cifrar datos debe utilizarse también para descifrar datos.

**EXTENSIÓN SOAP.** Una extensión SOAP es un mecanismo de extensibilidad admitido por ASP.NET que permite extender el procesamiento de mensajes de SOAP. La extensión SOAP permite inspeccionar o modificar un mensaje en determinadas etapas del ciclo de procesamiento en el cliente o el servidor.

**FICHERO.** Un fichero es la unidad mínima de almacenamiento de información. Los archivos son un tipo de ficheros, es decir, son ficheros que pueden albergar otros ficheros. En general, archivo y fichero se consideran sinónimos, a excepción del entorno Windows, donde a los ficheros se les denomina archivos, es decir, todo documento en Windows se almacena en un archivo. Sin embargo, en este entorno se denomina Fichero a una utilidad incluida que es una sencilla base de datos de dos campos.

**FIRMA DIGITAL.** La firma digital se utiliza en la autenticación de mensajes para garantizar la validez del emisor del mensaje, la integridad del mensaje y la invariabilidad de los datos durante el tránsito. La firma de los datos no supone la alteración de los mismos, sino únicamente la creación de una cadena de firma digital que se transmite con los datos. Las firmas digitales se crean con algoritmos de firma de clave pública como el cifrado de clave pública RSA.

**FIRMA DIGITAL XML.** Una firma digital XML es una firma digital que se aplica a un documento XML.

**FRAMEWORK.** Forma abreviada para referirse a la biblioteca de clases framework (framework class library), que es el nombre colectivo para los cientos de clases que componen al .NET Framework. Los servicios proporcionados por el FCL incluyen funcionalidad de tiempo de ejecución de núcleo (tipos básicos y colecciones, E/S a nivel de archivo y red, acceso a servicios de sistema, etc.), interacción con bases de datos, consumo y producción de XML, soporte para la construcción de aplicaciones cliente basadas en Web y aplicaciones cliente de escritorio y servicios Web basados en XML y SOAP.

**FTP.** File Transfer Protocol. Protocolo de Transferencia de Archivos. Uno de los protocolos de transferencia de ficheros mas usado en Internet.

**FUNCIÓN.** En programación, una sección nombrada de un programa que realiza una tarea específica. En este sentido, una función es un tipo de procedimiento o rutina. Algunos lenguajes de programación hacen la distinción entre una función, que regresa un valor, y un procedimiento, que realiza alguna operación pero no regresa ningún valor.

La mayoría de los lenguajes de programación vienen con un conjunto de funciones pre-escritas que son mantenidas en una librería; también es posible escribir funciones propias para realizar tareas específicas. El término función también se utiliza como sinónimo de operación y comando; por ejemplo se ejecuta la función borrar para eliminar una palabra.

**GUI.** Siglas de Graphical User Interface: interfaz gráfica de usuario.

**HACKER.** Experto en informática capaz de entrar en sistemas cuyo acceso es restringido. No necesariamente con malas intenciones.

**HARDWARE.** Es el término que indica todas las partes físicas, eléctricas y mecánicas de una computadora. Significa literalmente "partes duras" y se emplea en contraposición al término software, que significa "partes blandas", es decir, los programas de una computadora. A los componentes que es posible ver y tocar se les llama en jerga computacional "hardware", palabra inglesa cuyo significado es máquina o "cosa dura".

**HASH.** Un hash es un valor numérico de longitud fija que identifica datos de forma unívoca. Los valores

hash se utilizan para comprobar la integridad de los datos que se envían a través de canales no seguros. Puede compararse el valor hash de los datos recibidos con el valor hash de los datos que se enviaron para determinar si se alteraron los datos. Los valores hash también se utilizan en las firmas digitales. Dado que se pueden utilizar valores hash pequeños para representar cantidades de datos de mayor tamaño, sólo es necesario firmar el hash de un mensaje, en lugar de todos los datos del mismo.

**HEADER.** Cabecera. Primera parte de un paquete de datos que contiene información sobre las características de este.

**HERRAMIENTA.** Término aplicado a un programa que desarrolla servicios específicos.

**HIPERTEXTO.** Programa de generación de documentos con un sistema de acceso que puede jerarquizar el mismo usuario que crea el documento. Los documentos creados con un programa de este tipo han sido habitualmente orientados a su utilización en multimedia, debido a su asombrosa versatilidad.

**HOST.** Anfitrión. Computador conectado a Internet. Computador en general.

**HTML.** Hyper Text Markup Language. Lenguaje de Marcas de Hipertexto. Lenguaje para elaborar páginas Web actualmente se encuentra en su versión 3. Fue desarrollado en el CERN (Conseil Europeen pour la Recherche Nucleaire. Consejo Europeo para la Investigación Nuclear).

**HTTP.** Iniciales de HyperText Transfer Protocol (Protocolo de transferencias de hipertexto). Protocolo usado por la red World Wide Web, http define como se les da formato a los mensajes, como son transmitidos y que acciones deben tomar los servidores Web y los navegadores en respuesta a varios comandos. Por ejemplo, al introducir un URL en el navegador, este manda un comando http al servidor Web ordenando a transmitir la página Web solicitada. http es llamado un protocolo sin estado porque cada comando es ejecutado independientemente, sin conocimiento de los comandos que vienen atrás de el. Esta es la principal razón de que sea difícil implementar sitios Web que reaccionen inteligentemente a entradas de usuario. Esto puede solucionarse empleando diversas tecnologías nuevas como ActiveX, Javascript y cookies.

**HTTPS.** URL creada por Netscape Communications Corporation para designar documentos que llegan desde un servidor WWWseguro. Esta seguridad es dada por el protocolo SSL (Secure Sockets Layer) basado en la tecnología de encriptación y autentificación desarrollada por la RSA Data Security Inc.

**IDENTIDAD.** La identidad se refiere a una característica de un usuario o servicio que puede identificarlo de forma unívoca. Por ejemplo, a menudo es un nombre para mostrar que suele tener la forma "autoridad/nombre de usuario".

**IETF.** Internet Engineering Task Force (Fuerza de Trabajo de Ingeniería de Internet.)

**IMPLEMENTACIÓN.** Es una forma de llevar a la práctica un determinado concepto de diseño bajo unas ciertas circunstancias.

**INFORMÁTICA.** Contracción de INFORmación autoMÁTICA. Es un campo de conocimientos que abarca todos los aspectos relacionados con computadoras y con el tratamiento automático de la información.

**INICIO DE SESIÓN.** Un inicio de sesión define el contexto de seguridad en cada proceso que ejecuta. Cuando se inicia una sesión en un equipo de forma interactiva, se crea un inicio de sesión interactivo para alojar el Shell de Windows y cualquier proceso que se inicie de forma interactiva. Cuando un proceso se conecta en nombre de un usuario a un equipo remoto, se utilizan las credenciales del usuario (almacenadas en la caché del inicio de sesión local) para controlar las solicitudes de autenticación del equipo remoto. Si el proceso de autenticación es correcto, se establece un inicio de sesión de red en el equipo remoto, que representa el trabajo realizado en nombre del usuario en el equipo remoto.

**INTEGRIDAD.** Los canales de comunicación segura deben garantizar además la protección de los datos frente a modificaciones accidentales o deliberadas (malintencionadas) durante el tránsito. La integridad suelen proporcionarla los códigos de autenticación de mensajes (MAC).

**INTERNET.** Conjunto de redes y ruteadores que utilizan el protocolo TCP/IP y que funciona como una sola gran red. Es la red de redes. Nacida como experimento del ministerio de defensa americano.

**INTRANET.** Se llaman así a las redes tipo Internet pero que son de uso interno, por ejemplo, la red corporativa de una empresa que utilizara protocolo TCP/IP y servicios similares como WWW.

**IP.** Internet Protocol. Protocolo de Internet. Es un protocolo de bajo nivel para redes que describe la manera cómo el usuario puede comunicarse con los miembros Internet. Bajo este se agrupan los protocolos de Internet. También se refiere a las direcciones de red Internet.

**IPSEC.** (SEGURIDAD DEL PROTOCOLO INTERNET). IPSec es una forma de seguridad en el transporte. IPSec se diseñó para cifrar datos mientras se transmiten entre dos equipos para evitar que sean modificados o interpretados.

**ISO.** International Standard Organization. Organización Internacional de Estándares.

**ISP.** Internet Service Provider. Proveedor de Servicios Internet.

**JAVA.** Lenguaje de programación orientado a objeto parecido al C++. Usado en WWW para la telecarga y telejecucion de programas en el computador cliente. Desarrollado por Sun microsystems, con el propósito de mejorar las capacidades de las páginas de Web. Los programas en JAVA son llamados Applets.

**JAVASCRIPT.** Formalmente llamado LiveScript, este lenguaje fue desarrollado por Netscape. Concebido después del JAVA; su principal diferencia radica en que el programa se halla embebido en un archivo HTML, en lugar de ser un ejecutable que se carga cuando se carga una página de Web.

**J2EE.** Java 2 Enterprise Edition. (Edición Empresarial). J2EE es un ambiente independiente de la plataforma, basado en Java de Sun para el desarrollo, construcción y distribución de aplicaciones en línea basadas en Web. J2EE consiste en un conjunto de servicios, API's y protocolos que proveen la funcionalidad para desarrollar aplicaciones multicapas basadas en Web.

**KERBEROS.** Kerberos es un protocolo de autenticación que admiten los sistemas operativos Windows 2000 y posterior. Kerberos admite la forma ampliada de suplantación denominada delegación, que permite que el contexto de seguridad de un llamador tenga acceso a los recursos de la red y a los recursos locales del sistema operativo del servidor.

**LAN.** Local Area Network. Red de Area Local. Una red de área local es un sistema de comunicación de alta velocidad de transmisión. Estos sistemas están diseñados para permitir la comunicación y transmisión de datos entre estaciones de trabajo inteligentes, comúnmente conocidas como Computadoras Personales. Todas las PCs, conectadas a una red local, pueden enviar y recibir información. Como su mismo nombre lo indica, una red local es un sistema que cubre distancias cortas. Una red local se limita a una planta o un edificio.

**LDAP** (PROTOCOLO LIGERO DE ACCESO A DIRECTORIOS). LDAP es un protocolo que se utiliza para tener acceso a servicios de directorio como Active Directory.

**LIBRERÍA.** Una colección de archivos, en programación una librería es una colección de rutinas precompiladas que puede usar un programa. Las rutinas, algunas veces llamadas módulos, son almacenadas en formato de objeto. Las librerías son particularmente útiles para almacenar rutinas usadas frecuentemente porque no necesitan ligarse explícitamente a cada programa que las usa. El enlazador automáticamente busca las rutinas en las librerías que no encuentra en ningún otro lado. En ambientes Windows, los archivos de librerías tienen la extensión .DLL

**LINK.** Enlace. Unión. Hiperenlace. Se llama así a las partes de una página WEB que nos llevan a otra parte de la misma o nos enlaza con otro servidor.

**LISTA DE CONTROL DE ACCESO (ACL).** Una lista de control de acceso (ACL) es una lista ordenada de entradas de control de acceso (ACE) adjuntas a un objeto asegurable. El sistema operativo Windows utiliza dos tipos de ACL: una lista de control de acceso discrecional (DACL) que se utiliza para especificar los derechos de acceso de un usuario o grupo de usuarios y una lista de control de acceso al sistema (SACL) que se utiliza para determinar cuándo deben generar mensajes de auditoria los tipos de acceso específicos.

**LISTA DE REVOCACIONES DE CERTIFICADOS (CRL).** Una lista CRL es un documento que mantiene y publica una entidad emisora de certificados (CA) y que contiene los certificados emitidos por la CA que ya no son válidos.

**LOGIN.** Entrada de identificación, conexión. Igual que logon.

**LSA** (AUTORIDAD DE SEGURIDAD LOCAL). La Autoridad de seguridad local (LSA) es un subsistema de Windows local que proporciona servicios de autenticación.

**MAC** (CÓDIGO DE AUTENTICACIÓN DE MENSAJES). El código de autenticación de mensajes es un valor hash que se agrega a un mensaje para proporcionar integridad. Cuando se utiliza un algoritmo MAC para generar un Hash, la aplicación receptora debe tener también la clave de la sesión para volver a calcular el valor hash y comprobar que los datos del mensaje no se han modificado.

**MAIL.** El correo electrónico es el servicio más básico, antiguo, y más utilizado dentro de Internet. La mensajería electrónica es el medio más eficaz y más rápido de comunicación, permite intercambiar además de mensajes, programas, audio, video e imágenes.

**MAINFRAME.** Término inglés empleado para designar computadoras de grandes dimensiones.

**MAN.** Metropolitan Area Network. Red de Area Metropolitana.

**MEGABYTE.** (MB): unidad de medida de una memoria. 1 megabyte = 1024 kilobytes = 1.048.576 bytes.

**MEMORIA.** Se designa de este modo a un área de almacenamiento de una computadora que contiene datos e instrucciones.

**MEMORIA.** Aquella parte de un sistema computador, a menudo un sistema de almacenamiento a base de núcleos magnéticos, que almacena el programa y los datos en proceso y que proporciona un acceso rápido y directo a ella. Algunas veces, se le denomina "memoria principal" para distinguirla de los sistemas auxiliares de almacenamiento.

**MÉTODO.** En programación orientada a objetos, es un procedimiento que es ejecutado cuando un objeto recibe un mensaje. Un método es realmente lo mismo que un procedimiento, función o rutina en lenguajes de programación procedurales. La única diferencia es que en los lenguajes orientados a objetos, un método siempre se asocia con una clase.

**MICROPROCESADOR.** Es un componente electrónico de una computadora, también llamado circuito integrado o chip, que contiene las partes fundamentales de una computadora y los circuitos necesarios para que la computadora lleve a cabo sus cálculos.

**MÓDULO.** Término referido a componentes de un programa o sistema que se pueden identificar por separado y a los que es posible dirigirse también separadamente. Es equivalente al término segmento.

**MULTIMEDIA.** Tratamiento informático avanzado de las tecnologías más recientes de sonido e imagen que engloba e integra funciones como la animación gráfica, la manipulación y digitalización de imágenes y sonido.

**MULTIPROCESO.** Sistema en el que se utilizan varios procesadores funcionando simultáneamente y compartiendo tanto las memorias centrales como las auxiliares y los periféricos.

**NTLM.** NTLM (Windows NT LAN Manager) es un protocolo de autenticación desafío/respuesta que se utiliza en redes con sistemas que ejecutan versiones del sistema operativo Microsoft Windows NT anteriores a Windows 2000 y en sistemas independientes.

**OBJETO.** Por lo general, cualquier cosa que puede ser manipulada y seleccionada individualmente. Esta puede incluir figuras e imágenes que aparecen en la pantalla o bien entidades de software menos tangibles. En la programación orientada a objetos, por ejemplo, un objeto es una entidad auto-contenida que consiste tanto en los datos como en los procedimientos para manipular los datos.

**ODBC.** Siglas de Open Database Connectivity: Conectividad Abierta de Bases de Datos - estándares que simplifican el acceso a diferentes DBMSs de forma transparente garantizan el acceso a los datos de bases, posiblemente remotas, de distintas compañías.

**ORM.** Modelado de Objetos Relacionado.

**OSI.** Open Systems Interconnection. Interconexión de Sistemas Abiertos. Modelo de referencia de interconexión de sistemas abiertos propuesto por la ISO. Divide las tareas de la red en siete niveles.

**PACKET.** Paquete Cantidad mínima de datos que se transmite en una red o entre dispositivos. Tiene una estructura y longitud distinta según el protocolo al que pertenezca. También llamado TRAMA.

**PAP.** Password Authentication Protocol. Protocolo de Autentificacion por Password. Protocolo que permite al sistema verificar la identidad del otro punto de la conexión mediante password.

**PARTICIÓN DE DISCO.** Es una zona de un disco duro formada por áreas de memoria consecutivas; y separada de forma lógica de otras áreas que también lo forman.

**POSIX.** Estándar universal que define cómo debe ser un sistema operativo de "tipo UNIX" y que especifica una serie de normas para operación de las aplicaciones que se ejecutan en éste sistema operativo.

**PAR DE CLAVES.** Un par de claves es un par formado por una clave pública y una privada que pertenecen a una entidad y se utilizan para cifrar y descifrar datos.

**PDA.** Iniciales para Personal Digital Assistant, Asistente Digital Personal, un dispositivo de mano que combina características de cómputo, teléfono/fax, red e Internet. Un PDA típico puede funcionar como teléfono celular, fax, navegador Web y organizador personal. A diferencia de las computadoras portátiles, la mayoría de los PDA comenzaron basados en pluma, es decir usando un bolígrafo como dispositivo de entrada en vez de un teclado. Esto significa que también incluyen características de reconocimiento de escritura. Algunos PDA también pueden reaccionar a entradas por voz utilizando tecnologías de reconocimiento de voz. Los PDA actuales están disponibles ya sea en versión con teclado o con dispositivo señalador (pluma). Los PDA también son llamados palmtops, computadoras de mano y computadoras de bolsillo.

**PERFIL DE USUARIO.** Los perfiles de usuario contienen la información de configuración de un usuario. Esta información incluye la organización del escritorio, los grupos de programas personales, los elementos de programa, los colores de pantalla, los protectores de pantalla, las conexiones de red, etc. Cuando un usuario inicia una sesión de forma interactiva, el sistema carga el perfil del usuario y configura el entorno según la información del perfil.

**PERMISO.** Un permiso es una regla asociada a un objeto para regular cual o cuales usuarios obtienen acceso a dicho objeto y en que manera. Los permisos son asignados por el administrador del sistema, por el dueño del objeto o por el usuario que tenga permisos de control total sobre el objeto.

**PERMISO NTFS.** Un permiso utilizado en exclusivamente en particiones o volúmenes formateados usando el sistema NTFS. Son utilizados para controlar el acceso que un usuario, grupo o aplicación tiene a un archivo u objeto. Existen 5 tipos de permisos NTFS:

Lectura (Read), Escritura (Write), Lectura y Ejecución (Read & Execute), Modificar (Modify) y Control Total (Full Control)

**PKCS.** (estándares de criptografía de clave pública). PKCS es un conjunto de estándares de sintaxis para la criptografía de clave pública que abarca funciones de seguridad como, por ejemplo, los métodos para firmar datos, intercambiar claves, solicitar certificados, y cifrar y descifrar claves públicas.

**PRINCIPIO DE PRIVILEGIO MÍNIMO.** El principio de privilegio mínimo es el concepto de ejecutar código ejecutable con la identidad del proceso más débil posible. De esta forma se limitan los posibles daños producidos en caso de que la seguridad del proceso se viera comprometida. Si un usuario malintencionado logra insertar código en un proceso del servidor, los privilegios concedidos a dicho proceso determinan en gran medida los tipos de operaciones que puede ejecutar dicho usuario.

**PRIVACIDAD.** La privacidad consiste en garantizar la confidencialidad de los datos para que no puedan ser visualizados por usuarios malintencionados con software de supervisión de la red. La privacidad suele proporcionarse mediante el cifrado.

**PRIVILEGIO.** Un privilegio es el derecho de un usuario de ejecutar diversas operaciones relacionadas con el sistema como, por ejemplo, cerrar el sistema, cargar controladores de dispositivos o cambiar la hora del sistema. El testigo de acceso de un usuario contiene una lista de privilegios del usuario o del grupo de usuarios.

**PROCESADOR.** La parte central de un sistema computador que proporciona y controla las funciones aritméticas, lógicas y de transferencias requeridas para comparar, mover, calcular y, de cualquier manera, manipular y procesar datos.

**QL.** Siglas de Query Language. Un lenguaje de consulta.

**QUERY.** Una consulta -query- se define como una expresión lógica sobre los objetos y relaciones definidos en el esquema conceptual; el resultado es la identificación de un subconjunto lógico de la base de datos.

**RAID.** Iniciales de Redundant Array of Independent (or Inexpensive) Disks (Arreglo Redundante de Discos Independientes), una categoría de unidades de discos que emplean combinaciones de 2 o más unidades para obtener tolerancia a fallos y rendimiento. Los discos RAID son usados generalmente en servidores. Existen varios niveles de RAID los tres más comunes son 0, 3 y 5.

Nivel 0: Provee partición de los datos (repartiendo bloques de cada archivo en múltiples discos) pero no ofrece redundancia. Esto mejora el rendimiento reduciendo los tiempos de acceso a datos pero no ofrece tolerancia a fallos. Si un disco falla, se pierde la información.

Nivel 1: Provee espejeo (mirroring) de discos, lo que se escribe en un disco, se escribe en otro al mismo tiempo, ofreciendo así tolerancia a fallos pero disminuye el rendimiento.

Nivel 3: Lo mismo que en nivel 0, pero también reserva un disco dedicado para corrección de errores de datos. Provee un buen desempeño y un cierto nivel de tolerancia a fallos.

Nivel 5: Provee partición de datos a nivel de byte y también particiona información para la corrección de datos. Lo que resulta de un excelente desempeño y buena tolerancia a fallos.

**RAM.** Random Access Memory. Memoria de Acceso Aleatorio. Varios son los tipos de memoria que se usa en las computadoras. La más conocida son las RAM. Se les llama así porque es posible dirigirse directamente a la célula donde se encuentra almacenada la información. Su principal característica es que la información se almacena en ellas provisoriamente, pudiendo ser grabadas una y otra vez, al igual que un casete de sonido. La memoria RAM se puede comparar a un escritorio, donde se coloca los papeles con que se va a trabajar. Mientras más grande el escritorio más papeles soporta simultáneamente para ser procesados.

**RDBMS.** Siglas de Relational Database Management System: Sistemas Manejadores de Bases de Datos Relacionales.

**ROOT.** Raiz. En sistemas de ficheros se refiere al directorio raiz. En Unix se refiere al usuario principal.

**ROUTER.** Dispositivo conectado a dos o más redes que se encarga únicamente de tareas de comunicaciones.

**RSA.** RSA Data Security, Inc., es uno de los principales desarrolladores y publicadores de estándares de criptografía de clave pública. El nombre RSA responde a las iniciales de los nombres de los tres desarrolladores y propietarios de la empresa: Rivest, Shamir y Adleman.

**SACL** (lista de control de acceso al sistema)
Una SACL se asocia a un objeto asegurable (mediante un descriptor de seguridad) y especifica los tipos de operaciones ejecutadas por usuarios concretos que deberían generar mensajes de auditoria.

**SEGURIDAD DE ACCESO AL CÓDIGO.** La seguridad de acceso al código es una forma de seguridad de .NET que se utiliza para controlar el acceso del código a los recursos protegidos.

**SERVIDOR.** Computadora o dispositivo en una red que administra los recursos de red. Por ejemplo, un servidor de archivos en una computadora dedicada para almacenar archivos, cualquier usuario en la red puede almacenar archivos en el servidor. Un servidor de impresión es una computadora que administra una o más impresoras y un servidor de red es una computadora que administra el tráfico de red. Un servidor de base de datos es sistema computacional que procesa consultas de base de datos. Los

servidores son generalmente dedicados, es decir que no realizan otras tareas más que sus tareas como servidor. Sin embargo, en sistemas operativos multiproceso una sola computadora puede ejecutar varios programas a la vez. Un servidor en este caso se puede referir al programa que administra los recursos más que a la computadora en si.

**SERVIDOR DE APLICACIONES.** Un servidor de aplicaciones es un equipo servidor dedicado, independiente de un servidor Web cliente. El servidor de aplicaciones suele alojar servicios Web, componentes remotos o aplicaciones de Servicios Empresariales que contienen la mayoría de la lógica empresarial de una aplicación.

**SERVLET.** Tecnología de la plataforma Java para mejorar y extender a los servidores web. Los servlets proveen métodos basados en componentes e independientes de la plataforma para construir aplicaciones basadas en Web sin las limitaciones de desempeño de los programas basados en CGI. A diferencia de los mecanismos de extensión de servidores propietarios (por ejemplo Server API de Netscape o módulos de Apache) los servlets son independientes de la plataforma y de los servidores.

**SHA.** (algoritmo hash seguro) SHA es un algoritmo que se utiliza para generar un valor hash o implícito. El algoritmo SHA original ha sido reemplazado por el algoritmo SHA1 mejorado.

**SHELL.** Es un procedimiento mediante el cual se puede acceder temporalmente al sistema operativo desde el interior de un programa. En Windows es una ventana de aplicación especial que permite lanzar otras aplicaciones.

**SID.** (Identificador de seguridad) Un identificador de seguridad (SID) identifica de forma unívoca a un usuario o a un grupo de usuarios en un dominio. Un SID es un valor de longitud variable formado por un nivel de revisión, un valor de la autoridad que autentica (el emisor del SID, que suele ser Windows), un conjunto de valores de subautoridad (normalmente el dominio de la red) y un identificador relativo (RID), que es único en la combinación de autoridad/subautoridad que autentica. Los SID nunca se reutilizan, incluso cuando se elimina una cuenta de usuario y se vuelve a crear con la misma combinación de nombre y contraseña

**SNIFFER.** Literalmente "Husmeador". Pequeño programa que busca una cadena numérica o de caracteres en los paquetes que atraviesan un nodo con objeto de conseguir alguna información. Normalmente su uso es ilegal.

**SOAP.** Simple Object Access Protocol. (Protocolo de acceso a objetos simples) SOAP es un formato específico de representación de llamadas a procedimientos remotos basado en XML y que viaja sobre HTTP. Es un protocolo para intercambiar información en un entorno distribuido. Lo utilizan los servicios Web.

**SOFTWARE.** Esta palabra inglesa que significa "cosa suave", tiene dos significados. (a) Uno amplio, de "procedimientos lógicos, para la cooperación armónica de un grupo de personas y máquinas, persiguiendo un objetivo común"; (b) el otro restringido, de "programas de computadora", o conjunto de instrucciones, que se pone en la memoria de una computadora para dirigir sus operaciones. Es un conjunto de instrucciones que cargadas en el hardware de una computadora hacen que este pueda funcionar y realizar tareas. Puede traducirse en castellano como "partes blandas" y es el término contrario a HARDWARE, "partes duras".

**SSL.** Secure Sockets Layer. Capa de Socket Segura. Protocolo que ofrece funciones de seguridad a nivel de la capa de transporte para TCP.

**TCB.** (base de computación de confianza) Una TCB es un límite que define la parte de un sistema en la que se confía para exigir la directiva de seguridad. El código ejecutable que se ejecuta en la TCB puede realizar operaciones sin ser objeto de las comprobaciones de seguridad habituales. Los controladores de dispositivos se ejecutan en la TCB. El código de usuario se ejecuta en la TCB si la cuenta del proceso asociado tiene el privilegio "Actuar como parte del sistema operativo". El código de usuario que se ejecuta en la cuenta SISTEMA local también se ejecuta en los límites de la TCB.

**TCP/IP.** Transmission Control Protocol / Internet Protocol. El término describe dos mecanismos de software empleados para posibilitar la múltiple comunicación entre computadoras de manera libre de error. TCP/IP es el lenguaje común de la Internet, el que permite que diferentes tipos de computadoras utilicen la red y comuniquen unas con otras, indiferentemente de la plataforma o sistema operativo que usen.

**TELNET.** Protocolo y aplicaciones que permiten conexión como terminal remota a una computadora anfitriona, en una localización remota.

**TESTIGO DE ACCESO.** Un testigo de acceso es una estructura de datos adjunta a cada proceso de Windows. Mantiene información del contexto de seguridad del proceso, que incluye un SID de usuario que identifica el principal que representa al inicio de sesión y los atributos de autorización, como los privilegios y los SID del grupo del usuario. Cada testigo de acceso se asocia exactamente con un inicio de sesión, mientras que un inicio de sesión puede contener varios testigos de acceso, uno por cada proceso que se inicie en la sesión y, opcionalmente, otros testigos del subproceso adjuntos a subprocesos individuales.

**TESTIGO DEL SUBPROCESO.** Un testigo del subproceso es un testigo de acceso temporal asociado a un determinado subproceso. Cuando se crea un subproceso, no tiene testigo de acceso y las operaciones seguras que realiza el subproceso utilizan la información obtenida del testigo del proceso. Una situación típica en la que un subproceso adquiere un testigo de acceso es cuando un subproceso de un proceso servidor desea trabajar en nombre de un cliente. En tal situación, el subproceso representa al cliente mediante la adquisición de un testigo de acceso. Los testigos del proceso también se denominan testigos temporales y testigos de suplantación.

**TEXTO CIFRADO.** El texto cifrado son datos que se han cifrado.

**TEXTO SIN CIFRAR.** El texto sin cifrar son datos que no se han cifrado.

**UDDI.** Universal Description, Discovery and Integration. (Descripción Universal, Descubrimiento e Integración). Un directorio distribuido basado en Web que habilita los negocios para enlistarse ellos mismos en Internet y descubrirse unos a otros, similar a la tradicional sección amarilla o directorio blanco.

**UML.** Siglas de Unified Modeling Language. Es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

**UNC.** Universal Naming Convention, Convención de nombrado Universal. Forma de nombrar a los recursos dentro de una red. Primero se indica el nombre del servidor, la compartición, el directorio y al ultimo el archivo o recurso. Ejemplo: \\SERVIDOR\Compartición\ MyFolder\archivo.txt

**URL.** Universal Resource Locator. Nombre genérico de la dirección en Internet, Indica al usuario dónde localizar un archivo HTML determinado, en la Web.

**UTILIDAD.** Programa que desempeña una tarea específica de uso general. Forma parte del software del sistema.

**VALOR SALT.** El valor salt son datos aleatorios que pueden utilizarse junto con los datos cifrados o hash para aumentar las defensas necesarias para proteger los datos de ataques de diccionario por fuerza bruta. Se suele colocar delante de los datos cifrados o hash.

**WAN.** Siglas de Wide Area Network (red de área global). Es una red de computadoras heterogénea sin limitación de distancia en la que sus componentes pueden estar conectados de muy diversos modos, no solamente mediante cables.

**WEB.** Site. Sitio en el World Wide Web. Conjunto de páginas Web que forman una unidad de presentación, como una revista o libro. Un sitio está formado por una colección de páginas Web.

**WSDL.** Web Service Description Language. (Lenguaje de Descripción de Servicios Web), un lenguaje con formato XML usado para describir las capacidades de un servicio Web como colecciones de comunicación capaces de intercambiar mensajes. WSDL es una parte integral de UDDI, un registro de negocios a nivel mundial basado en XML.
WSDL es el lenguaje que usado por UDDI. WSDL fue desarrollado en conjunto entre Microsoft e IBM.

**WWW.** (World Wide Web). Servidor de información, desarrollado en el CERN (Laboratorio Europeo de Física de Partículas), buscando construir un sistema distribuido hipermedia e hipertexto. También llamado WEB y W3. Existen gran cantidad de clientes WWW para diferentes plataformas.

**XML.** Iniciales para eXtensible Markup Language (Lenguaje de marcas extendido), una especificación desarrollada por el W3C. XML es una versión de SGML, diseñada especialmente para documentos Web. Permite a los diseñadores crear sus propias etiquetas hechas a la medida, permitiendo la definición, transmisión, validación e interpretación de los datos entre aplicaciones y entre organizaciones.

**.NET** Una plataforma de sistema operativo de Microsoft que incorpora aplicaciones, una suite de herramientas y servicios y un cambio en la infraestructura de la estrategia Web de la compañía. Desde el punto de vista del usuario existen 4 puntos principales de .NET:
Se eliminan las fronteras entre las aplicaciones e Internet. En vez de interactuar con alguna aplicación o con un solo sitio Web, .NET conectará al usuario a un arreglo de computadoras y servicios que intercambiaran y combinaran objetos y datos.
El Software será rentado como un servicio hospedado en Internet en vez de se comprado en una tienda. Esencialmente, Internet hospedará todas las aplicaciones y todos los datos.
Los usuarios tendrán acceso a su información en Internet desde cualquier dispositivo, en cualquier tiempo y desde cualquier parte del mundo.
Habrá nuevas formas de interactuar con los datos de las aplicaciones, tales como el reconocimiento de voz y de escritura a mano.
.NET depende de 4 estándares de Internet:
HTTP
XML
SOAP
UDDI
Microsoft ve esta nueva tecnología como revolucionaria, permitiendo a los usuarios de Internet hacer cosas que no eran posibles antes, tales como integrar fax, correo electrónico y servicios telefónicos, centralizar el almacenamiento de datos y sincronizar todos los dispositivos computacionales para ser automáticamente actualizados.

# BIBLIOGRAFIA

# Bibliografía


Farley, Marc. Lan Times (1998). "**Guía de Seguridad e Integración de Datos**". Editorial McGraw Hill, México.

I. Jacobson, G. Booch, J. Rumbaugh. (2000). "**El Proceso Unificado de Desarrollo de Software**". Editorial Pearson Educación, S.A., Madrid España.

I. Jacobson, G. Booch, J. Rumbaugh. (1999). "**El Lenguaje Unificado de Modelado**". Editorial Addison Wesley Iberoamericana, S.A., Madrid España.

Larman, Graig, (1999). "**UML y Patrones. Introducción al análisis y diseño orientado a objetos**". Editorial Pretince Hall, México.

Kercher Jeff (2002) "**Autenticación en ASP .NET: Directrices de seguridad para .NET**" Editorial Microsoft Corporation.

Lawrance Pfleeger, Shari (1991). "**Software Engineering**", Editorial Macmillan Publishing Company, 2nd edition.

Ortal Robert (Fecha) "Client-**Server"Programming Witw**" Editorial. Van Nostrand Reinltold, Extended Edition.

Pattini, Mario y Del Peso, Emilio (1998) "**Auditoria Informática: Un Enfoque Práctico**".Editorial. Alfa Omega S.A. de C.V. México.

Pratt, Terrence W (Fecha) "**Programming languages: design and implementation**" 4th edition, Prentice Hall.

Pressman, Roger S. (1998). "**Ingenieria de Software, un enfoque practico**", Editorial McGraw-Hill,4a edicion.

Sommerville, Ian (2001). "**Software Engineering**" Editorial Addison-Wesley Publishers.

Silberschatz, Korth, Sudarshan, (2002). "**Fundamentos de Bases de Datos**" Editorial McGraw-Hill,4a edicion.

Tanenbaum, Andrew (1997). "**Redes de Computadoras**" Tercera Edición, Editorial Prentice Hall, México.

Vallabhanenis, Rao (1989). "**Auditing Computer Security. Manual with Case Studies**". Editorial. Wiley, U.S.A..

Yager Thomas (2001) "**Guía de desarrollo de aplicaciones Web con Windows 2000**"Editorial Prentice Hall, Madrid, pp. 21, 25-258

Yann, Derrien (1998). "**Técnicas de Auditoría Informática.**". Editorial Alfa Omega S.A. de C.V., México.

**CÓDIGO PENAL PARA EL ESTADO DE SINALOA**
(Ref. por Decreto número 270, publicado en el P. O. No. 038 del 28 de Marzo de 2003)

Barry M. Leiner, Vinton G. Cerf. (et al) "**Una breve historia de Internet (Primera Parte)**",

http://www.ati.es/DOCS/internet/histint/histint1.html

Carlos C. Tapang, "**Explicación del lenguaje WSDL (Web Services Description Language)**"
http://www.microsoft.com/spanish/msdn/articulos/archivo/091101/voices/wsdlexplained.asp

Carlos Lizárraga Celaya, "**Servicios Web**"
http://www.fisica.uson.mx/carlos/WebServices/WSRevolution.htm

Consol García, "**EVOLUCIÓN EN EL ACCESO A BASES DE DATOS Y VALORES AÑADIDOS**",
http://fesabid98.florida-uni.es/Comunicaciones/c_garcia.htm

Eve Andersson, "**Arquitectura de sistemas distribuidos**",
http://www.galileo.edu/wp/display/2213/

Jorge Espinosa, "**Introducción a n-Capas con VFP y VB**",
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art20.asp

Jorge Oblitas, "**Administración de excepciones en ASP.NET**",
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art23.asp

Jorge Oblitas y Jorge Serrano Pérez, "Los **eventos en el nuevo Global.asax**"
http://www.microsoft.com spanish msdn/comunidad/mtj.net/voices/art24.asp

Rob Caron, "**Introducción a servicios Web XML en Visual Studio .NET**",
http://www.microsoft.com/spanish/msdn/articulos/archivo/120402/voices/vbtchgettingstartedwithxmlwebservicesinvisualstudionet.asp

Rocio López L., "**Auditoría de Sistemas de Información**", http://whatis.techtarget.com/, 2001

Scott Seely, "**Seguridad HTTP y servicios Web de ASP.NET**",
http://www.microsoft.com/spanish/msdn/articulos/archivo/111002/voices/httpsecurity.asp,
Agosto de 2002.

"**Biblioteca - artículos electrónicos**",
http://www.tribunalmmm.gob.mx/biblioteca/almadelia/indice.htm

"**Cliente servidor**", http://ar.geocities.com/r_niella/Document/t_cap1.htm

"Comprobar **la identidad de Windows en una aplicación de cliente**",
http://es.gotdotnet.com/quickstart/howto/doc/security/WindowsIdentityCheck.aspx

"**Curso de SQL**", http://www.aulaclic.org/sql/f_sql.htm

"**Diagrama de Flujo de Datos (DFD)**"
http://fismat.umich.mx/~emurguia/mipagina/tesis/node43.html

"**EL ABC DE LA PROPIEDAD INTELECTUAL**", http://www.bufete-evf.com.mx/html/espanol/abc.htm

"**Microsoft .NET Framework FAQ**",
http://msdn.microsoft.com/netframework/using/gettingstarted/default.aspx

**"Programa Estratégico para Protección de Técnología de Microsoft"**,
http://www.microsoft.com/latam/seguridad mstpp.asp

**"¿Qué es Microsoft .NET?"**
http://www.microsoft.com/latam/netframework producto/resumen.asp

"Uso **del analista empresarial: Modelado de funciones de objetos con Visual Studio.NET**",
http://www.microsoft.com/spanish msdn articulos archivo/140901/voices orm.asp

"**Security**", http://www.webopedia.com Networks Security

**"Servicios Web y .NET"**
http://www.microsoft.com/spanish/msdn articulos/archivo/041002/voices/datati.asp

"**UML**", http://loghog.corc.uni.edu.ni/~jorge upoli_uml/refs/UML.html