



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DESARROLLO DE MÓDULO DINÁMICO
DE ADMINISTRACIÓN DE BASES DE DATOS
CON SOFTWARE LIBRE**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTAN:

CHÁVEZ SALOMÉ RUTH MARIBEL

MÉNDEZ LÓPEZ ROSENDO

DIRECTOR DE TESIS:

ING. ALBERTO GONZÁLEZ GUIZAR

MÉXICO D. F. 2007





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

A la Universidad Nacional Autónoma de México por haber permitido nuestro desarrollo profesional.

A nuestros padres, por el apoyo que hemos recibido de ellos, así como de sus consejos, conocimientos y paciencia que de ellos obtuvimos.

Al Ing. Gonzáles Guízar por habernos orientado en el desarrollo de la misma.

A nuestros sinodales por su apoyo y por las observaciones y tiempo dedicado.

ÍNDICE

<i>ÍNDICE DE FIGURAS</i>	3
<i>ÍNDICE DE TABLAS</i>	3
1 INTRODUCCIÓN	4
1.1 Antecedentes	7
1.2 Definición del problema	15
1.3 Objetivo	16
1.4 Relevancia	16
1.5 Resultados esperados	16
2 ANÁLISIS DEL PROBLEMA	17
2.1 Conceptos básicos de bases de datos	19
2.2 Funciones de los DBMS	26
2.3 Conectividad	27
2.4 Arquitectura de los DBMS	29
2.5 Estándares ANSI-SQL	31
2.5.1 SQL-89	32
2.5.2 SQL-92	33
2.5.3 SQL 3.....	34
2.6 Tareas y funciones del administrador de la base de datos (DBA)	38
3 ANÁLISIS DE HERRAMIENTAS USADAS PARA EL DESARROLLO DEL SISTEMA	42
3.1 Sistemas Administradores de Base de Datos Relacionales (RDBMS)	44
3.2 Servidor Web	51
3.3 Herramientas de programación	57
3.3.1 Tecnología PHP	58
3.3.2 Tecnología ASP.....	59
3.3.3 Tecnología JSP	61
3.3.4 ColdFusion	62
3.3.5 Elección de la Herramienta.....	64
3.3.6 Requerimientos de seguridad informática que debe de cubrir el sistema.....	65
4 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	66
4.1 Antecedentes	67
4.2 Diseño e implementación del sistema	73
5 CONCLUSIONES	92
6 GLOSARIO	95
7 BIBLIOGRAFÍA	103

ÍNDICE DE FIGURAS

FIGURA 1. DIAGRAMA DE FUNCIONAMIENTO DE LA TECNOLOGÍA CGI	10
FIGURA 2. DIAGRAMA DE FUNCIONAMIENTO DE LAS PÁGINAS ESTÁTICAS	11
FIGURA 3. DIAGRAMA DE FUNCIONAMIENTO DE LAS PÁGINAS DINÁMICAS	12
FIGURA 4. DIAGRAMA DE FUNCIONAMIENTO DE LA TECNOLOGÍA LAMP	13
FIGURA 5. INTERACCIÓN DE USUARIOS CON BASE DE DATOS	20
FIGURA 6. COMPONENTES DE UNA BASE DE DATOS	23
FIGURA 7. INTERACCIÓN DEL USUARIO CON LA BASE DE DATOS.....	26
FIGURA 8. ARQUITECTURA SEPARADA DE DBMS.....	29
FIGURA 9. ARQUITECTURA INTEGRADA DE DBMS.....	30
FIGURA 10. ARQUITECTURA DE ANSI/SPARC	31
FIGURA 11. INTERACTIVIDAD DE USUARIO CON BASES DE DATOS VÍA WEB	43
FIGURA 12. ARQUITECTURA DE APLICACIONES WEB REPRESENTADA EN ESQUEMA MULTINIVEL	53
FIGURA 13. ARQUITECTURA DE APLICACIONES WEB REPRESENTADA EN ESQUEMA DE TRES NIVELES	54
FIGURA 14. FUNCIONAMIENTO DE COLDFUSION.....	64
FIGURA 15. SUCESIÓN DE FASES	68
FIGURA 16. ESQUEMA GENERAL DE OPERACIÓN DE UNA FASE	69
FIGURA 17. EJEMPLO DE CICLO LINEAL PARA UN PROYECTO DE CONSTRUCCIÓN	70
FIGURA 18. CICLO DE VIDA EN ESPIRAL.....	71
FIGURA 19. CICLO DE VIDA PARA EL MODULO DE ADMINISTRACIÓN DE BASES DE DATOS	73
FIGURA 20. RED GENERAL DE DESARROLLO DEL PROYECTO	74
FIGURA 21. COMPONENTES DEL SISTEMA.....	75
FIGURA 22. ACCESO AL SISTEMA	76
FIGURA 23. COMPORTAMIENTO GENERAL DEL SISTEMA.....	77
FIGURA 24. SECCIÓN DE ADMINISTRACIÓN DEL SISTEMA	77
FIGURA 25. PROCEDIMIENTO PARA DAR DE ALTA BASE DE DATOS	78
FIGURA 26. SECCIÓN DE MONITOREO DEL SISTEMA	79
FIGURA 27. SECCIÓN DE SEGURIDAD DEL SISTEMA.....	79
FIGURA 28. FUNCIONAMIENTO DEL MÓDULO DE SERVIDORES.....	80
FIGURA 29. FUNCIONAMIENTO DEL MÓDULO DE USUARIO	81
FIGURA 30. FUNCIONAMIENTO DEL MODULO DE RDBMS	82
FIGURA 31. PLANTILLA DE LAS PÁGINAS WEB.....	83
FIGURA 32. PLANTILLA DE LA PÁGINA PRINCIPAL	84
FIGURA 33. PLANTILLA PARA LOS MÓDULOS DE ADMINISTRACIÓN MONITOREO Y SEGURIDAD	84
FIGURA 34. DIAGRAMA DE CLASES.....	85
FIGURA 35. PÁGINA PRINCIPAL DEL SISTEMA.....	86
FIGURA 36. ADMINISTRACIÓN DE BASES DE DATOS	87
FIGURA 37. ADMINISTRADOR DE RDBMS	87
FIGURA 38. PANEL DE ADMINSTRACIÓN DEL SISTEMA	88
FIGURA 39. DIAGRAMA DE PAQUETES DEL SISTEMA	89

ÍNDICE DE TABLAS

TABLA 1. ADMINISTRADORES DE BASES DE DATOS POPULARES.....	6
TABLA 2. RDBMS MÁS POPULARES.....	50
TABLA 3. SERVIDORES WEB EXISTENTES	57



1 INTRODUCCIÓN

En este capítulo se describen brevemente algunas herramientas que administran bases de datos, las tecnologías Web, los tipos de páginas Web que hay y se plantean los problemas de la administración de las bases de datos, así como el objetivo, la relevancia y los resultados que se esperan obtener del presente trabajo de tesis.

Introducción

Desde los principios de la humanidad la información adquirió una importancia trascendental al ser requerida por el hombre para permitirle desarrollar diversas actividades. La necesidad de controlar el uso y generación de la información se hizo presente por el acelerado ritmo con que era recopilada, almacenada, procesada y transmitida. Así pues, fue ineludible el desarrollo de bases de datos que permitieran administrar el constante crecimiento de los sistemas de información, asegurando una estructuración eficaz de los datos.

Dentro de los usos y ventajas que ofrecen las bases de datos, tenemos que facilitan el almacenamiento de grandes cantidades de información, permiten la recuperación flexible y rápida de información, permiten organizar y reorganizar la información, así como imprimir o distribuir la información de numerosas formas.

La información ha sido definida como un “conjunto de datos interrelacionados entre sí, que tienen un significado del cual se puede obtener elementos para la toma de decisiones”, por lo que una base de datos es considerada como una colección de datos interrelacionados, almacenados más o menos de manera permanentemente en una computadora. Dichos datos son compartidos por diferentes usuarios y programas de aplicación, quienes no conocen los detalles de la estructura de almacenamiento

Una base de datos permite el manejo de gran volumen de información de manera tal que ésta pueda encontrarse íntegra, disponible y manipulable, siempre que un usuario autorizado lo solicite. Debido a esto es de vital importancia contar con las herramientas necesarias para administrar de manera eficiente y segura una base de datos, para ello podemos hacer uso de programas y aplicaciones que nos permitan manipular libremente la información de una base de datos.

Un sistema de administración de bases de datos es el software que controla la organización, almacenamiento, recuperación, seguridad e integridad y manejo de la información en una base de datos haciendo uso de algún modelo de datos. Acepta pedidos de datos desde un programa de aplicación o cliente y le ordena al sistema operativo transferir los datos apropiados.

El sistema de administración de bases de datos relaciona la base de datos física con los usuarios, maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios tomando en consideración la seguridad de la información, por tanto, protege a los usuarios de la base de datos contra los pormenores a nivel de hardware. Ofrece una vista de la base de datos que está por encima del nivel de hardware y apoya las operaciones del usuario que se expresan en términos de esa vista de nivel superior.

Con el auge del Internet las bases de datos comenzaron a tener una importancia mayúscula al ser usadas para el envío, localización y recepción de

información, mediante el manejo dinámico y flexible de los datos. Al principio los sitios Web se componían de un conjunto de archivos almacenados independientemente, lo cual ocasionaba graves problemas de administración especialmente tratándose de un sitio extenso, ya que es difícil mantener actualizados y enlazados numerosos documentos separados.

Conjugando el poder de convocatoria de Internet con las bases de datos, se tiene la posibilidad de unificar y simplificar el acceso a la información, pues no existe entre los usuarios y las paginas Web la limitación del sistema operativo, permitiéndole a cualquier persona el uso de cualquier plataforma para acceder a páginas Web (que pueden ser desplegadas con un navegador de Internet que funciona en cualquier plataforma) con servidores de bases de datos alojados en otra plataforma, sin tener que cambiar el formato o la estructura de la información dentro de las bases de datos.

Además la tarea de actualizar la información de un sitio Web disminuye drásticamente con el uso de las bases de datos, ya que los administradores tienen mayor control al automatizar el mantenimiento de bases de datos de forma remota. Al mismo tiempo, la unión de Internet con bases de datos permite que la información sea accesible desde cualquier parte del mundo, agrega mayor capacidad de almacenamiento de datos y ofrece accesibilidad al soportar distintas estructuras de los datos

En la actualidad existen diferentes herramientas que se pueden utilizar para administrar bases de datos vía Web, algunos ejemplos de estas aplicaciones son los siguientes:

APLICACIÓN	CARACTERÍSTICAS
PhpMyAdmin	Proyecto de código abierto desarrollado en PHP para administrar vía Web una base de datos MySQL. Código gratuito.
JavaMyAdmin	Interfase desarrollada en Java2 para la administración de bases de datos MySQL. Tiene las mismas funciones que PhpMyAdmin pero con menos problemas de seguridad. Código gratuito.
PhpPgAdmin	Administrador vía Web para acceder a servidores de bases de datos PostgreSQL. Es un programa válido para funciones básicas y algunas avanzadas como procedimientos o vistas. Código gratuito.
MySQL-Front	Entorno gráfico (similar a Windows) para el manejo y administración de bases de datos MySQL a partir de la versión 3.23. No está programado en PHP y HTML, por lo que la respuesta es mucho más rápida y eficaz. Si el ISP prohíbe la conexión a la Base de Datos desde el exterior, dispone de una interfase de conexión basada en PHP. Código gratuito.

Tabla 1. Administradores de bases de datos populares

Como se puede notar a pesar de ser aplicaciones gratuitas, todas ellas tienen la desventaja de estar ligadas únicamente a un sistema administrador de bases de datos específico, por lo que nuestro principal interés es desarrollar un módulo dinámico que permita a muchos usuarios manipular muchos tipos de sistemas administradores de bases de datos vía Web.

1.1 Antecedentes

El origen de la popularidad de Internet se remonta al año 1989, cuando Tim Berners-Lee, investigador de los laboratorios suizos del CERN (Organización Europea de Investigaciones Nucleares) desarrolló un sistema de intercambio de datos basado en el uso de documentos de texto (llamados posteriormente páginas Web) enlazados entre sí por hipervínculos y escritos en un subconjunto sencillo del complejo lenguaje SGML¹ llamado HTML². Debido a que la audiencia de Internet aumentaba, se fue mejorando el lenguaje HTML inicial hasta llegar a su versión actual, o a XHTML³, que es el resultado de describirlo siguiendo las reglas del lenguaje estándar de marcas conocido como XML⁴. La desventaja que mostraba este sistema radicaba en que las páginas HTML eran simples archivos colocados en un servidor Web que los enviaba a los clientes tal cual, sin posibilidad de interacción con los mismos y obligando a que la actualización de dichos archivos se realizara editando su contenido en el servidor. Así pues, conforme la red fue creciendo también crecieron las necesidades de los clientes por lo cual se fue introduciendo en el lenguaje y en el funcionamiento de los servidores Web diversos mecanismos que permitirían la modificación y generación dinámica del contenido de las páginas en Internet.

Tecnologías en el lado del cliente

Para conseguir aumentar el dinamismo de los sitios Web se le otorgo al lenguaje HTML la capacidad de poder incluir scripts de código escritos en diferentes lenguajes que los navegadores interpretarían para, a través de ellos, modificar el aspecto de las páginas e interactuar con el navegador. A esto se le conoce como DHTML⁵ y forma parte de las denominadas **tecnologías en el lado del cliente**, pues requiere de un soporte directo por parte del navegador del cliente. Ahí es precisamente donde radica su principal problema: los fabricantes de

¹ SGML: Standardised Generalised Markup Language, Lenguaje de Mercado Generalizado Estandarizado.

² HTML: Hyper Text Markup Language, Lenguaje de Marcas para Hiper Texto.

³ XHTML: EXTended HTML, HTML extendido.

⁴ XML: Extensible Markup Language, Lenguaje de Mercado Extensible.

⁵ DHTML: Dynamic HTML, HTML Dinámico.

navegadores nunca han llegado a un acuerdo a la hora de implementar esto, y cada uno soporta tanto diferentes lenguajes de scripts como diversas capacidades dentro de un mismo lenguaje, lo que hace que sea difícil la creación y mantenimiento de páginas DHTML compatibles con los navegadores más utilizados.

Ejemplos de tecnologías en el lado del cliente son:

❖ Estáticas

- HTML
- CSS ⁶
- JavaScript
- XHTML, XML

❖ Dinámicas

- DHTML
- VBScript
- Java Applets
- Controles Active X (Visual C++, Visual Basic)
- Flash

Las tecnologías dinámicas del lado del cliente se basan en las capacidades del navegador para poder interpretar el contenido dinámico que le envía el servidor y mostrarlo correctamente en una PC. Esta idea puede parecer buena, porque el servidor deja en manos del cliente gran parte del trabajo de procesamiento del contenido. Pero esto también significa que generalmente se debe disponer de la última versión del navegador favorito del cliente y una PC lo suficientemente rápida para poder hacer el trabajo que le delega el servidor.

Tecnologías en el lado del servidor

Las denominadas **tecnologías en el lado del servidor** se basan en la ejecución de código en el servidor que se encarga de generar dinámicamente el contenido a enviar a cada cliente. Esto se realiza de forma completamente transparente al cliente, quien recibe los contenidos tal como si fuesen archivos estáticos almacenados en el servidor, por lo que no necesita ninguna clase de soporte especial para tratarlos. Aunque se puede argumentar que esta opción provoca un aumento en la carga de trabajo del servidor, es cierto que el alto nivel

⁶ CSS: Cascading Style Sheets, Hojas de estilo en cascada.

de compatibilidad que proporciona está haciendo que se convierta en la tendencia actual en el desarrollo de aplicaciones en el ciberespacio, especialmente debido a la proliferación de todo tipo de dispositivos con acceso a Internet y con capacidades de visualización y procesamiento múltiples, cuyos navegadores suelen ser muy simples y no admitir scripts.

La primera tecnología en el lado del servidor fue **CGI** ⁷ (Figura 1), que permitía a un programador escribir un programa ejecutable en prácticamente cualquier lenguaje (como C, Pascal, o los más utilizados lenguajes de script como Perl o Python) que era capaz de leer lo que enviaba un cliente desde un formulario, escribir en la página que era enviada al cliente, interactuando en el servidor antes, trabajando con archivos, enviando un correo electrónico, etc. En esta tecnología se define cómo pasaría el servidor la información (mediante la línea de comandos y ciertas variables de entorno) y cómo proporcionaría el servidor el contenido a enviar al cliente (mediante su salida estándar). Su gran problema es que cada vez que llegue una nueva petición **CGI** al servidor, éste tendrá que lanzar un nuevo ejecutable para tratarla y descargarlo de memoria tras ello, lo que resulta un proceso muy costoso y que tiende a degradar mucho el rendimiento y la escalabilidad de la aplicación Web.

Como solución a este problema se inició la tendencia de sustituir los programas **CGI** por librerías que el servidor enlazara dinámicamente en su mismo espacio de memoria, cuyos tiempos de carga y ejecución fueran muy inferiores que los de programas que se ejecutan en espacios de memoria independientes, sin ser necesario descargarlas tras atender a cada cliente. Entre los ejemplos de este tipo de tecnología tenemos:

- La tecnología llamada **ISAPI** ⁸ de Microsoft, admitida por su servidor **IIS** ⁹ y por servidores de otras empresas. Sin embargo, la implementación de estas librerías tiene el problema de ser relativamente complicada, requiriendo de personal especialmente calificado para ello.
- La tecnología llamada **NSAPI** ¹⁰ de Netscape Communications, admitida por el servidor Netscape FastTrack y otros servidores Netscape.

⁷ CGI: Common Gateway Interface, Interfaz de entrada común

⁸ ISAPI: Internet Server API, API para servidores de Internet

⁹ IIS: Internet Information Server, Servidor de información de Internet

¹⁰ NSAPI: Netscape Server API, API para servidores Netscape



Figura 1. Diagrama de funcionamiento de la tecnología CGI

Ejemplos de tecnologías en el lado del servidor son:

❖ Dinámicas:

- ASP ¹¹
- JSP ¹²
- Scripts de servidor (Perl, Shell Unix)
- PHP ¹³
- Programación CGI (C++, C, VBasic)
- API ¹⁴ Web propietaria (ISAPI, NSAPI)

Como el contenido del sitio dinámico es procesado en el servidor de acuerdo a los pedidos del cliente, el navegador (y en consecuencia la PC) no realiza ningún trabajo de interpretación. Esto significa que se puede usar prácticamente cualquier navegador, en cualquier plataforma, con cualquier PC para acceder a este tipo de sitios.

¹¹ ASP: Active Server Pages, Páginas de Servidor Activo

¹² JSP: Java Server Pages, Páginas de Servidor Java

¹³ PHP: Hypertext Preprocessor, Preprocesador de hipertexto

¹⁴ API: Application Programming Interface, Interfase para la Programación de Aplicaciones

Tipos de páginas Web

Los tipos de páginas Web podemos dividirlos en dos grandes grupos por su tipo de ejecución: las que se ejecutan completamente en nuestro navegador llamadas páginas estáticas (Figura 2) y las que parte del código es generado en el momento que el usuario ingresa al sitio llamadas páginas dinámicas (Figura 3).

Las páginas estáticas solo ofrecen textos planos acompañados por imágenes y en ocasiones algunos contenidos multimedia (sonido o video). Por lo tanto, no ofrecen ninguna funcionalidad, más allá de los enlaces que pueden hacerse en ellas

Las páginas dinámicas son más versátiles y pueden realizarse en ellas presentaciones mas elaboradas. Por lo general cuentan con una base de datos, y estas páginas son fáciles de actualizar.

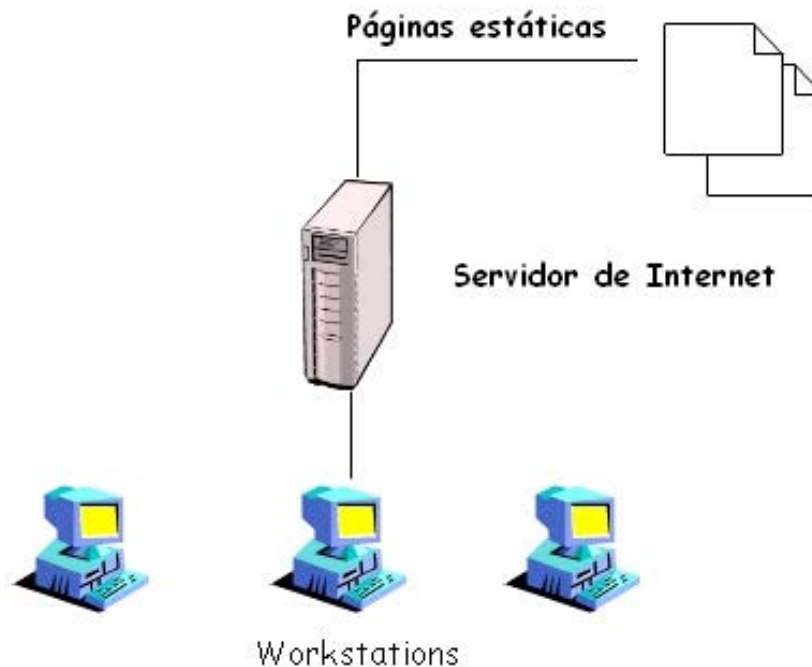


Figura 2. Diagrama de funcionamiento de las páginas estáticas

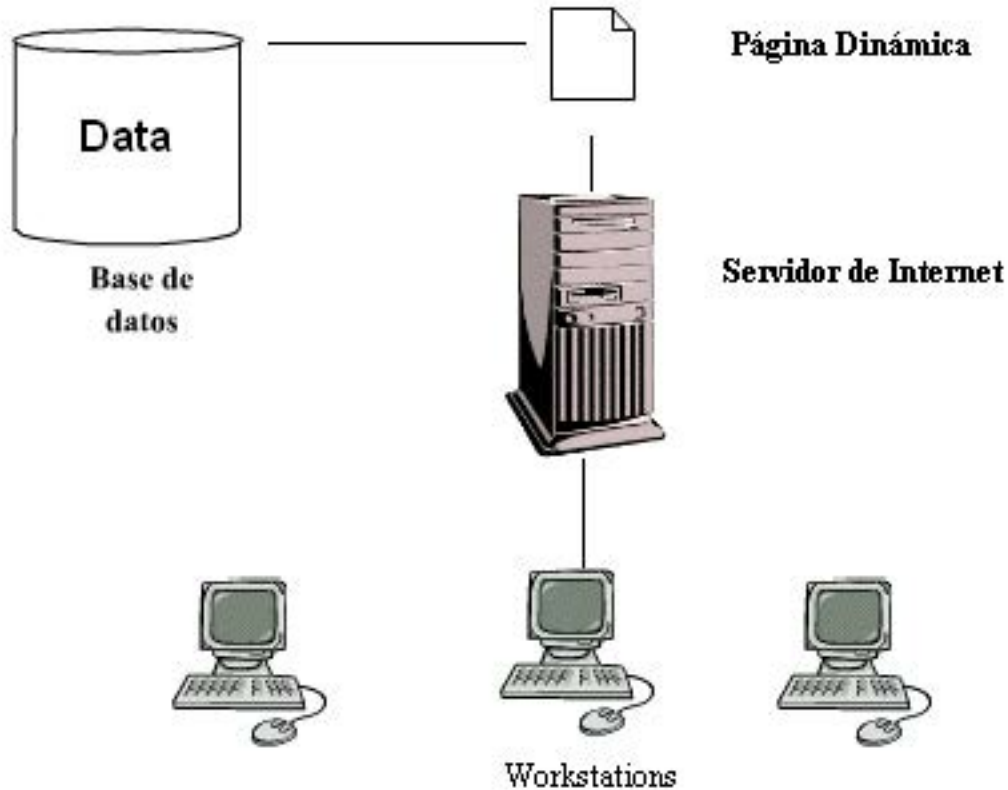


Figura 3. Diagrama de funcionamiento de las páginas dinámicas

Elementos de un proyecto Web

Entre las herramientas de cómputo más utilizadas actualmente encontramos a las bases de datos. Muchas veces en una página Web dinámica la base de datos es responsable de proporcionar la información que se mostrará al usuario, con la gran posibilidad de que la información pueda ser presentada de acuerdo al usuario que la solicite, de la fecha, la hora o dependiendo lo que se desee. La forma de mostrar los datos es indistinta, es posible tener diferentes plantillas que permitan desplegar los mismos datos, de manera que el usuario pueda ver la página como más le guste sin perder por ello nada de información, adaptando no sólo el diseño sino incluso definiendo que contenidos se pueden ver.

En el proceso de la planeación para crear una página Web dinámica se distinguen dos partes:

- ✓ **Primera parte:** Es la referente a la tecnología del servidor de Internet. Siendo más específicos, Microsoft cuenta con la plataforma .NET, la cual soporta diferentes lenguajes con la tecnología ASP, Macromedia tiene el servidor ColdFusion que permite trabajar con aplicaciones ASP, PHP y JSP, y en software libre se tiene Apache con PHP o Apache con Java y JSP.

- ✓ **Segunda parte:** Es el sistema administrador de bases de datos. Los mas conocidos son Oracle, PostgreSQL, Microsoft SQL Server o MySQL. Los servidores Windows comúnmente acostumbran usar Microsoft SQL Server, aunque también se usa MySQL como alternativa de bajo costo. En entornos Apache con programación en PHP la base de datos más utilizada, es MySQL. Una de las tecnologías con gran futuro en la red es la llamada LAMP (Linux - Apache - MySQL - PHP). En entornos de programación JSP (generalmente sobre Apache y Tomcat), se suele utilizar PostgreSQL o MySQL. Finalmente, una combinación clásica es la de Sun Solaris con iPlanet y Oracle, usando OC4J (Oracle Container for Java) como servidor de aplicaciones Java.

La tecnología LAMP (Figura 4) es gratuita, y Linux y Apache son posiblemente de lo mejor en sus correspondientes ámbitos. Sin embargo, MySQL tiene algunas limitaciones (no tiene algunas funcionalidades que sí tienen SQL Server o PostgreSQL), aunque resulta muy potente y útil para prácticamente cualquier tipo de aplicaciones. Y en cuanto a PHP, como lenguaje de programación es un lenguaje específico para Internet (a diferencia de los lenguajes ASP.NET) que cada vez está teniendo más aceptación, y del que podemos encontrar numerosas aplicaciones gratuitas fácilmente adaptables a nuestras necesidades.

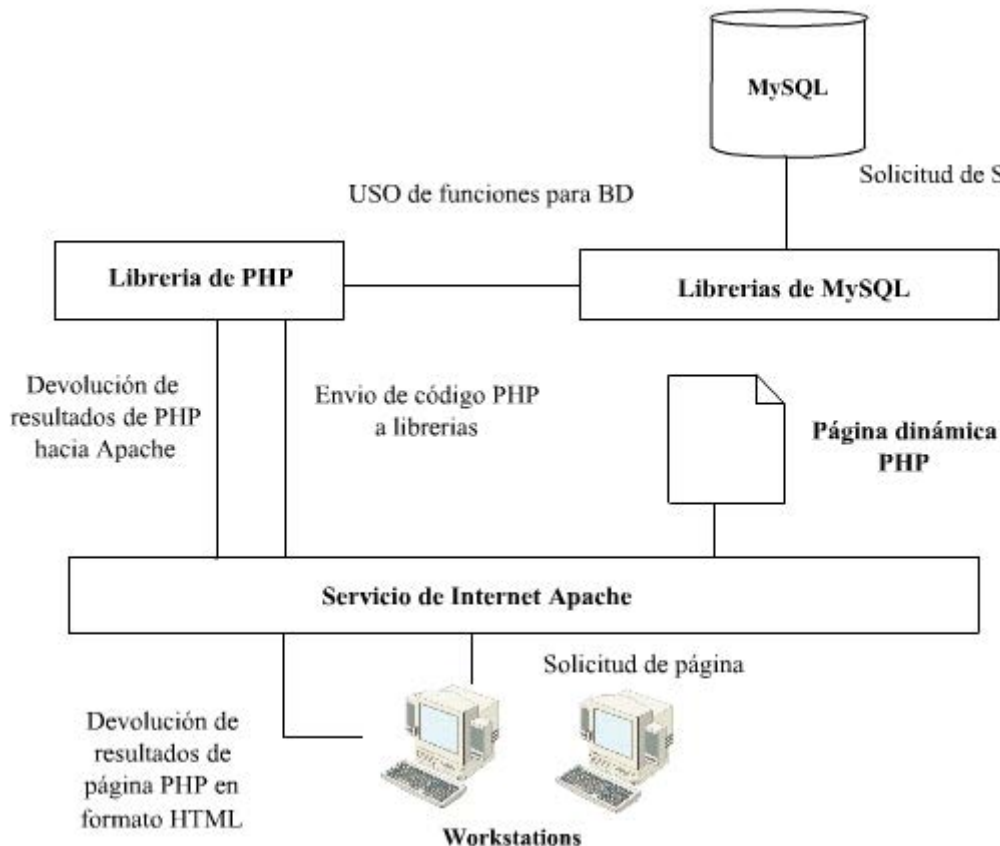


Figura 4. Diagrama de funcionamiento de la tecnología LAMP

En un proyecto Web pueden solicitarse páginas estáticas o dinámicas, de éstas pueden haber:

- Páginas estáticas en HTML (éstas pueden incluir imágenes, hoja de estilo, banners, scripts JavaScript, Visual Basic Script o Jscript, XML, y otras utilerías) las cuales se desarrollan rápidamente pues no suelen requerir mucho trabajo.
- Páginas estáticas en PHP, JSP, Java Servlets, ASP.Net, etc. (interactúan con servidores, cookies, archivos planos, o con otras herramientas) estas aplicaciones requieren más labor que las páginas estáticas con HTML, y de un mayor conocimiento de programación.
- Páginas dinámicas que interactúan con bases de datos y requieren páginas de administración de los sistemas. En este caso es en el que se invierte más tiempo, ya que se requieren altas, bajas y modificaciones de los registros de las bases de datos. Después, se hacen las páginas que se verán en el sitio en lo que con lleva el despliegue de valores de las bases de datos, y todo lo que se requiere para una buena presentación de las páginas. En todos estos procedimientos se requiere hacer pruebas para garantizar que no hay errores y que las páginas tienen la funcionalidad que el usuario requiere, éstas páginas son las mas laboriosas y en todas se ocupan las herramientas que se mencionaron anteriormente.

Para garantizar la calidad y el correcto funcionamiento de una página Web es necesario realizar pruebas. En el caso de las páginas estáticas hechas en HTML, se debe verificar que la página pueda verse completamente en varios navegadores (Internet Explorer, Netscape, Safari, etc.) sin presentar problemas, ya que los navegadores utilizan diferentes etiquetas HTML por lo que es posible que la página no se vea igual de un navegador a otro. Las páginas estáticas hechas en PHP, JSP, ASP.NET, etc. requieren de las mismas pruebas que las páginas en HTML y de pruebas de los scripts (verificando que no tienen errores de programación). Las pruebas en las páginas dinámicas contemplan todo lo de las páginas anteriores, también se verifica la correcta conexión con la base de datos y el script de administración.

Estas aplicaciones pueden ser utilizadas para personal de una empresa mediante una red interna (Intranet), para usuarios externos para promover a una empresa, con sus servicios, ofertas, etc.

En conclusión las páginas más adecuadas para cualquier tipo de aplicación son las páginas dinámicas, debido a que su contenido puede ser modificado con mayor facilidad que el de una página estática; ya que una página estática necesita ser modificada manualmente y si las páginas que necesitan ser modificadas son bastantes lleva mas tiempo modificar un solo dato, en cambio si son páginas dinámicas, con una sola modificación en una base de datos todas las páginas que utilizan ese dato son modificadas automáticamente y se ahorra mucho tiempo.

Las páginas dinámicas pueden ser utilizadas para administrar bases de datos, generar imágenes dinámicamente con lo que se pueden generar estadísticas, generar páginas que interactúen más con el usuario, interactuar con otras herramientas como administradores de listas de discusión, monitoreo de servidores, etc.

1.2 Definición del problema

Hoy por hoy es de gran importancia controlar la rapidez en el trabajo, especialmente el tiempo que se invierte para desarrollar soluciones eficientes, seguras, confiables y poco costosas. Con la aparición de Internet, se ha creado toda una cultura de intercambio de información y una consecuencia de ello es la aparición de los sitios Web. Dado el apogeo que ha tenido la creación de sitios Web, el compromiso que adquiere un Web master es el de esforzarse por realizar su trabajo con calidad y responsabilidad, recordando siempre que los clientes continuamente esperan un producto bien hecho, barato y realizado con rapidez.

Además de la creación del diseño de las páginas o sitios Web, se requiere también desarrollar una base de datos con páginas administrativas de tal manera que el usuario final pueda añadir, suprimir y modificar propiedades directamente y de manera segura.

En el desarrollo de sistemas a través del Web la interacción con diversos sistemas administradores de Base de datos hace necesario el tener que dar mantenimiento a cada una de ellas, lo cual vuelve más complicado el proceso de administración y también caro, tanto por el personal capacitado como por las herramientas administrativas que se necesitan.

En la actualidad es necesario contar con una herramienta específica para cada administrador de base de datos, aún para realizar las tareas más sencillas de administración. Este problema puede solucionarse al tener una aplicación vía Web que pueda ser usada desde cualquier máquina con un navegador, que simplificará el tiempo de instalación, configuración y realización de las tareas administrativas de la base de datos.

Así definimos que el objetivo de la tesis será desarrollar un módulo dinámico de administración de bases de datos vía Web con la ayuda de software libre.

1.3 Objetivo

Desarrollar un módulo dinámico de administración de base de datos mediante software libre que pueda interactuar con diferentes sistemas administradores de bases de datos relacionales vía Web.

1.4 Relevancia

Debemos recordar que además del diseño de las páginas Web, es necesario crear una base de datos con páginas Web administrativas para que el cliente pueda editar propiedades del sitio directamente.

La relevancia que adquiere la creación de un módulo dinámico de administración de base de datos se refleja en la reducción del tiempo de desarrollo de las páginas administrativas de un sitio Web, pues con este módulo las páginas administrativas serán creadas automáticamente. De esta manera, el usuario solo deberá dedicarse a los scripts de despliegue de información, de diseño y a los scripts que interactúen con otras aplicaciones Web.

Debido a que el diseño de la página o sitio Web es decidido por el usuario, el módulo dinámico de administración de bases de datos muestra la ventaja de poder acoplarse tanto a cualquier diseño como a cualquier RDBMS¹⁵.

1.5 Resultados esperados

El resultado que esperamos obtener es que la aplicación desarrollada pueda interactuar libremente con diferentes sistemas administradores de bases de datos relacionales (asegurando mantener la integridad y seguridad de la información manipulada), ofreciendo al usuario la facilidad de ingresar al sistema por medio de cualquier navegador Web.

¹⁵ RDBMS: Relational Data Base Manager System, Sistema Administrador de Bases de Datos Relacionales



2 ANÁLISIS DEL PROBLEMA

En este capítulo se describen los conceptos básicos de las bases de datos, la conectividad y la arquitectura de los DBMS, los estándares ANSI-SQL, así como la historia de SQL y las funciones y tareas de los DBA.

Análisis del problema

Conforme se han desarrollado las bases de datos ha surgido la problemática de administrarlas correctamente. Los usuarios siempre están buscando sistemas que les faciliten las operaciones sobre las bases de datos, tratando de disminuir el tiempo y esfuerzo requerido en el aprendizaje y capacitación del uso de los sistemas. Es tarea de los programadores desarrollar alternativas que permitan disponer de métodos eficientes para mantener el control adecuado sobre una base de datos, tanto para usuarios principiantes como para usuarios avanzados. Actualmente existen herramientas que ofrecen al usuario interfaces para controlar un DBMS ¹ de manera sencilla, sin embargo dichas herramientas se enfocan a la administración de un DBMS específico, por lo que es necesario adquirir diferentes aplicaciones para que un solo usuario pueda administrar diferentes DBMS. Se pueden considerar otras soluciones para este problema:

- Crear scripts en lenguaje SQL para insertar, borrar o actualizar los datos, que al ejecutarlos en un DBMS los datos serán actualizados, borrados o modificados. Estos scripts en lenguaje SQL son susceptibles a ser rehusados para realizar otros procedimientos en múltiples DBMS, solo es necesario modificar los datos. Una desventaja es que para la gente que no conoce SQL puede ser difícil el diseño de scripts y para la gente que lo conoce puede ser tedioso.
- Desarrollar un sistema con un lenguaje X. Una ventaja en este caso es que no se necesitan grandes conocimientos para administrar el sistema, solo conocimientos básicos. En este caso pueden ser utilizados dos tipos de lenguajes, uno es el de Tecnologías Web y otro es el de nivel usuario, el segundo necesita ser instalado en la máquina del usuario. La ventaja del lenguaje de Tecnologías Web es que solo se necesita un navegador y saber la dirección del sistema, tener acceso a la administración del sistema y puede actualizar el sistema desde cualquier parte ya sea en la oficina, en la casa o en un cibercafé, desde cualquier parte del mundo. La desventaja que puede tener este sistema está en la seguridad. Aunque el software de nivel de usuario también está propenso a fallas de seguridad.

Es evidente que al desarrollar un sistema administrador de múltiples DBMS combinado con tecnologías Web es más factible tener problemas de seguridad, por lo que se recomienda tener siempre actualizado el antivirus, tener nuestro sistema operativo actualizado para poder cubrir las vulnerabilidades y utilizar herramientas como los firewall, antispam, etc. Si queremos actualizar desde un

¹ DBMS: Data Base Management System, Sistema Administrador de Bases de Datos

cibercafé, casa o incluso de nuestra propia empresa, debemos de tener la seguridad de que la máquina que vamos a utilizar tiene un nivel mínimo de seguridad, ya que puede estar sujeta a virus o intrusos que desean comprometer la información o datos de suma importancia para la empresa; aunque no hay sistemas seguros, debemos de hacer lo posible por protegerlos. Debemos estar concientes del tipo de navegador que estamos utilizando, pues hay algunos que tienen muchas vulnerabilidades, por lo que esta herramienta necesita estar actualizada. Las consecuencias de un sistema con baja seguridad pueden ir desde copia de proyectos, hasta la pérdida total de la base de datos de nuestro sistema. Por eso debemos de tomar conciencia que la seguridad es de suma importancia.

Hay medidas de seguridad que podemos considerar para nuestro desarrollo como control de acceso a los usuarios, ya sea dar permisos de solo lectura, modificación, altas o borrado de datos. También podemos crear tablas para saber qué usuario modifica, borra o inserta ciertos datos, además de tener siempre respaldada nuestra base de datos, en caso de pérdida total o parcial y verificar a qué tipo de vulnerabilidades está sujeto nuestro DBMS y parcharlo si es necesario.

Lo que buscamos en este trabajo de tesis es administrar con un sólo módulo variados sistemas de administración de bases de datos relaciones (RDBMS) dinámicamente, y lo limitaremos a RDMBS libres, para hacerlo más económico a los usuarios que quieran adquirir el módulo. Con el desarrollo del sistema el usuario podrá utilizarlo para fines de sitios Web o para una administración de bases de datos exclusivas de su empresa o de su interés personal desde cualquier lugar. Utilizaremos un lenguaje de tecnología Web libre para poder hacer esto más económico para el usuario final.

2.1 Conceptos básicos de bases de datos

Una base de datos es en esencia una colección de archivos relacionados entre sí, de la cual los usuarios pueden extraer información sin considerar las fronteras de los archivos. Un objetivo importante de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo para que el sistema sea manejable, los datos se deben extraer eficientemente

Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema (Figura 5):

- **Nivel físico:** Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en como se almacenan los datos en los dispositivos de almacenamiento (por ejemplo, mediante señaladores o índices para el acceso aleatorio a los datos).

- **Nivel conceptual:** El siguiente nivel más alto de abstracción, describe que datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos. Consta de las siguientes definiciones:
 - **Definición de los datos:** Se describen el tipo de datos y la longitud de campo todos los elementos direccionables en la base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos y registros conceptuales (entidades).
 - **Relaciones entre datos:** Se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

En el nivel conceptual la base de datos aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario.

- **Nivel de visión:** Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la base de datos al usuario acreditado para verla. El sistema puede proporcionar muchas visiones para la misma base de datos.

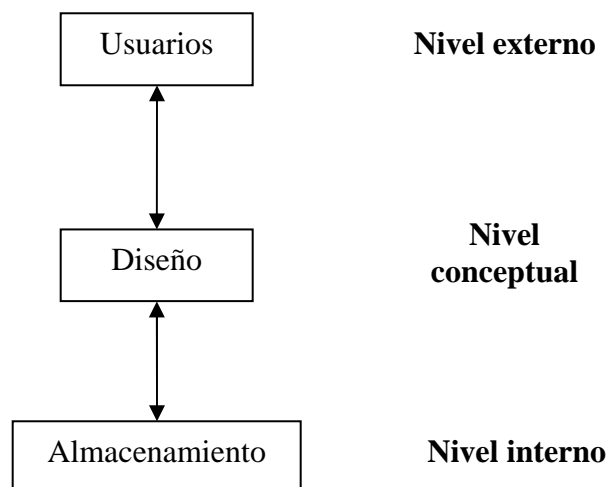


Figura 1. Interacción de usuarios con base de datos

Modelo de datos

Un modelo es una representación de la realidad que contiene las características generales de algo que se va a realizar. En base de datos, esta representación se elabora de forma gráfica. Un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Los modelos de datos se dividen en tres grupos:

- **Modelos lógicos basados en objetos:** Se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo representamos los datos de tal forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.
- **Modelos lógicos basados en registros:** Se utilizan para describir datos en los niveles conceptual y físico. Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de la implementación. Los tres modelos de datos más ampliamente aceptados son:
 - Modelo Relacional
 - Modelo de Red
 - Modelo Jerárquico
- **Modelos físicos de datos:** Se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos. Existen dos clasificaciones de este tipo que son:
 - Modelo unificador
 - Memoria de elementos.

Estructura general de un sistema de bases de datos

Un sistema de base de datos se encuentra dividido en módulos cada uno de los cuales controla una parte de la responsabilidad total de sistema. En la mayoría de los casos, el sistema operativo proporciona únicamente los servicios más básicos y el sistema de la base de datos debe partir de esa base y controlar además el manejo correcto de los datos. Así el diseño de un sistema de base de datos debe incluir la interfaz entre el sistema de base de datos y el sistema

operativo. Los componentes funcionales de un sistema de base de datos, son (figura 6):

- **Gestor de archivos:** Controla la asignación de espacio en la memoria del disco y de las estructuras de datos usadas para representar información
- **Manejador de base de datos:** Sirve de interfaz entre los datos y los programas de aplicación.
- **Procesador de consultas:** Traduce las proposiciones en lenguajes de consulta a instrucciones de bajo nivel. Además convierte la solicitud del usuario en una forma más eficiente.
- **Compilador de DDL:** Convierte las proposiciones DDL en un conjunto de tablas que contienen metadatos, estas se almacenan en el diccionario de datos.
- **Archivo de datos:** En él se encuentran almacenados físicamente los datos de una organización.
- **Diccionario de datos:** Contiene la información referente a la estructura de la base de datos.
- **Índices:** Permiten un rápido acceso a registros que contienen valores específicos.

Una forma gráfica de representar los componentes antes mencionados y la relación que existe entre ellos sería la siguiente (Figura 6).

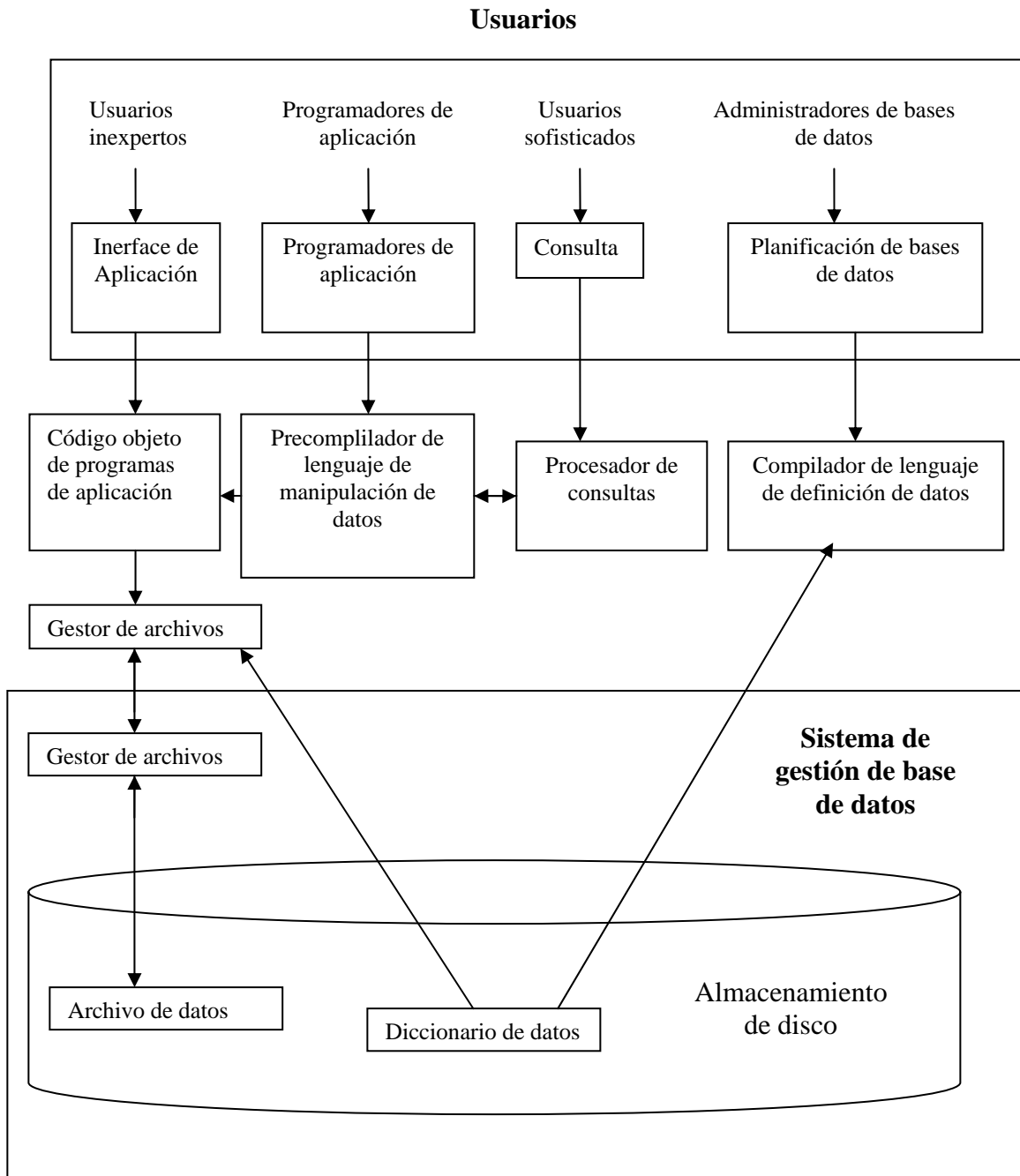


Figura 2. Componentes de una base de datos

Las funciones típicas de un DBMS son:

- **Lectura:** Deben permitir la lectura e incorporación de datos, independiente de la versión utilizada.

- **Manipulación y actualización:** Deben permitir atender las solicitudes del usuario para extraer, cambiar, adicionar y actualizar datos a la base de datos.
- **Independencia de datos:** Los sistemas actuales permiten aislar al usuario de la complejidad del almacenamiento físico de los datos.
- **Control de integridad:** No deben haber inconsistencias respecto a la definición y dominios en la estructura de las tablas.
- **Control de redundancias:** Minimizar los datos repetidos. Esto facilita la actualización ya que si hay datos repetidos habrá que modificarlos donde cada uno este. Si los datos repetidos no se actualizan simultáneamente se generarán inconsistencias y se dice que la base de datos estará corrupta.
- **Diccionario de datos:** Deben incluir una función de diccionario de datos donde se relacione información explicativa acerca de los datos con sus definiciones.
- **Metadatos:** Los datos sobre los datos son los metadatos. Corresponden a una gran ficha donde se describen la identificación, calidad, distribución y otra información relevante de un conjunto de datos.
- **Sincronización:** Una base de datos puede ser usada por dos usuarios al mismo tiempo y es necesario prevenir que se pueda modificar un mismo registro al simultáneamente. Cada usuario debe percibir en tiempo real los arreglos incorporados por el otro usuario.
- **Seguridad:** Se necesita un administrador (una persona, organización o software) que le brinde al sistema diferentes procesos para proteger la información y dar a los usuarios diferentes niveles de acceso.

Modelo de datos relacional

Codd propuso en 1970 que los sistemas de bases de datos deberían presentarse a los usuarios con una visión de los datos organizados en estructuras llamadas relaciones, definidas como conjuntos de tuplas (filas) y no como series o secuencias de objetos, con lo que el orden no es importante. Por tanto, detrás de una relación puede haber cualquier estructura de datos compleja que permita una respuesta rápida a una variedad de consultas. Codd hizo entonces énfasis en que el usuario de un sistema relacional sólo debía preocuparse por el qué consultar y no el cómo de las estructuras de almacenamiento (lo que ahora se conoce como modelo físico).

En 1985 Codd publicó sus famosas 12 reglas sobre el modelo relacional de bases de datos, un resumen de sus características fundamentales. Es preciso resaltar que todavía hoy algunas de estas reglas son de difícil implementación para los fabricantes de RDBMS. Además de ser considerado como el padre del modelo relacional, Codd también incursionó en el modelo multidimensional de análisis de datos conocido como OLAP (On Line Analytical Processing) y en 1993 Codd y algunos de sus colegas publicaron las “12 reglas para OLAP”.

La estructura fundamental del modelo relacional es la relación, es decir una tabla bidimensional constituida por tuplas (filas) y atributos (columnas). Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representan las propiedades de la entidad.

La base de datos relacional es muy flexible ya que los elementos que la integran se pueden ingresar de modo independiente a la estructura que quiera formarse con ellos (o sea, primero las tablas luego sus relaciones) mientras que en los otros modelos la estructura ya está definida. Se puede realizar cualquier clase de búsqueda de datos entre tablas siempre y cuando haya campos comunes entre estas.

La implementación de una base de datos relacional es algo costosa y el desempeño puede ser lento en algunos casos cuando el sistema deba realizar muchas confrontaciones entre tablas vinculadas. Aún así, es uno de los modelos de mayor uso y sólo son superadas por las bases de datos orientadas a objetos

La representación de un paisaje en bases de datos relacionales se ilustra mediante un MODELO ENTIDAD - RELACIÓN entendido como el conjunto de entidades, atributos y relaciones relevantes dentro de una base de datos:

- **Conjunto de entidades:** Los objetos relevantes para la base de datos.
- **Conjunto de atributos:** Las variables o características de los objetos. Cada atributo posee un conjunto de valores posibles, este conjunto se llama **DOMINIO**.
- **Conjunto de relaciones:** Los aspectos que permiten relacionar a cualquier nivel una entidad con otra.

Elaboración del modelo E-R:

1. Se definen las entidades relevantes
2. Se ilustran las entidades
3. Se reconocen que identificadores y atributos son propios de cada entidad
4. Se definen las relaciones

2.2 Funciones de los DBMS

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedidos rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el sistema administrador de archivos, a través de las sentencias en DML al comando del sistema de archivos. Así el Sistema administrador de base de datos es el responsable del verdadero almacenamiento de los datos.
- Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- Seguridad e integridad. Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

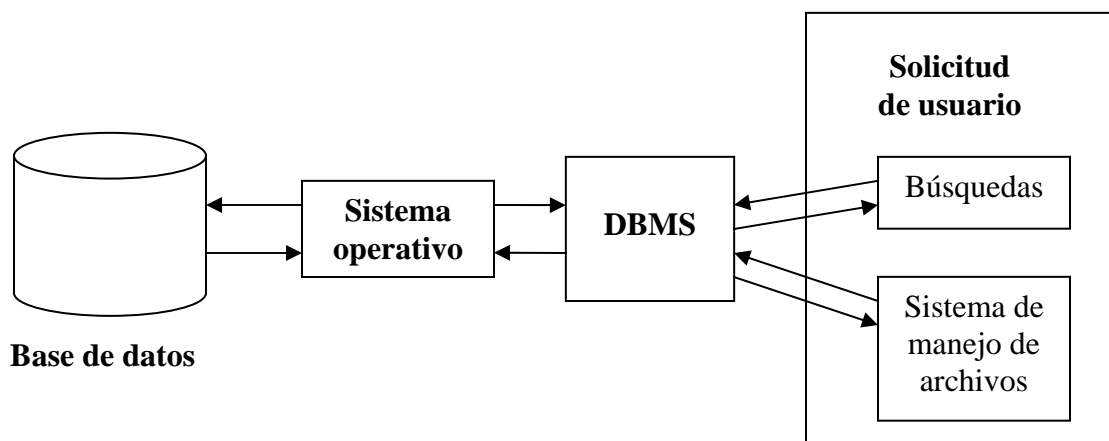


Figura 3. Interacción del usuario con la base de datos

La figura 7 muestra el DBMS como interfase entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

En sí, un sistema administrador de base de datos es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la puedan afectar.

2.3 Conectividad

Las API que se describen a continuación, son un claro ejemplo del proceso correspondiente a la conectividad de datos.

- **ODBC** ² : Esta tecnología proporciona una interfaz común para tener acceso a bases de datos SQL heterogéneas. ODBC está basado en SQL como un estándar para tener acceso a datos. ODBC permite la conexión fácil desde varios lenguajes de programación y se utiliza mucho en el entorno Windows. Sobre ODBC Microsoft ha construido sus extensiones OLE DB y ADO. Los ODBC se pueden clasificar en 3 categorías:
 - Los ODBC que permitan la realización de consultas y actualizaciones.
 - Los ODBC mediante los cuales se pueda llegar a la creación de tablas en la base de datos.
 - Los ODBC propios de los DBMS, los cuales se pueden llegar a manipular ciertas herramientas de administración.
- **CGI**: Es una de las soluciones que se está utilizando más para la creación de interfaces Web/DBMS. Entre las ventajas de la programación CGI, destaca la sencillez, ya que es muy fácil de entender, además de ser un lenguaje de programación independiente, ya que los escritos CGI pueden elaborarse en varios lenguajes. También es un estándar para usarse en todos los servidores Web, y funcionar bajo una arquitectura independiente, ya que ha sido creado para trabajar con cualquier arquitectura de servidor Web. Como la aplicación CGI se encuentra funcionando de forma independiente, no pone en peligro al servidor, en cuanto al cumplimiento de todas las tareas que éste se encuentre realizando, o al acceso del estado interno del mismo. Pero el CGI presenta cierta desventaja en su eficiencia, debido a que el servidor Web tiene que cargar el programa CGI y conectar y desconectar con la base de datos cada vez que se recibe una requisición. Además, no existe un registro del estado del servidor, sino que todo hay que hacerlo manualmente.
- **ISAPI**: Es la interfaz propuesta por Microsoft como una alternativa más rápida que el CGI, y está incluida en el Servidor Microsoft Internet

² ODBC: Open DataBase Connectivity, Conectividad abierta de bases de datos

Information (IIS). Así como los escritos CGI, los programas escritos usando ISAPI habilitan un usuario remoto para ejecutar un programa, busca información dentro de una base de datos, o intercambia información como otro software localizado en el servidor. Los programas escritos usando la interfaz ISAPI son compilados como bibliotecas de enlace dinámico (DLL - Dynamic Link Library), ya que son cargados por el servidor Web cuando éste se inicia. Dichos programas se vuelven residentes en memoria, por lo que se ejecutan mucho más rápido que las aplicaciones CGI, debido a que requieren menos tiempo de uso de CPU al no iniciar procesos separados. Uno de los programas ISAPI más usados es el HTTPODBC.DLL que se usa para enviar y/o devolver información hacia y desde las bases de datos, a través de ODBC. Además, ISAPI permite realizar un procesamiento previo de la solicitud y uno posterior de la respuesta, con lo cual manipula la solicitud/respuesta HTTP. Los filtros ISAPI pueden utilizarse para aplicaciones tales como autenticación, acceso o apertura de sesión.

- **NSPAI:** Es la API propuesta por Netscape para extender la funcionalidad de sus servidores.
- **DBI (PERL):** Perl es uno de los lenguajes más utilizados para programación en la Web y proporciona su propia interfaz de acceso a datos, llamada DBI (DataBase Interface). Es especialmente utilizado bajo plataformas Linux/Unix, solucionando las complejidades de ODBC en estos sistemas. DBI actúa como una abstracción para un conjunto de módulos DBD (DataBase Driver). Cada módulo DBD actúa como manejador de un sistema gestor de base de datos distinto. Existen módulos para prácticamente cualquier DBMS y puentes hacia otras tecnologías como ADO y JDBC.
- **JDBC (Java Data Base Connectivity):** Se trata del estándar para la conectividad entre el lenguaje Java y un amplio rango de sistemas gestores de bases de datos. Los JDBC pueden desenvolverse tanto en un nivel cliente, esto es, trabajando del lado de la aplicación, o en el servidor directamente relacionado con la base de datos. Cuando se encuentre a nivel cliente, trabajará con la tecnología ODBC para acceso a los datos. Hay diversos tipos de controladores JDBC:
 - El puente JDBC-ODBC: fue uno de los primeros controladores disponibles, implementa un enlace para utilizar un controlador ODBC desde Java. Con el tiempo han surgido controladores JDBC específicos para cada base de datos que mejoran el rendimiento del puente JDBC-ODBC.
 - Controladores Java parcialmente nativos: usan tanto código Java como binario específico de cada plataforma.

- Controladores JDBC-Net de Java puro: son controladores escritos completamente en Java que entienden un protocolo de red estándar (HTTP, etc.) y permiten comunicarse con un servidor de acceso a bases de datos, que es el que finalmente provee el acceso al SGBD específico (posiblemente con ODBC).
- Controladores de protocolo nativo en Java puro: escritos en Java puro, utilizan el protocolo específico de la marca del SGBD.
- **SQL LINKS:** Son controladores que se encargan de realizar la comunicación remota entre la aplicación y los servidores remotos de bases de datos, permitiendo una comunicación casi directa y muy rápida. Los ha desarrollado la empresa Inprise y permiten conexiones con otros servidores de bases de datos como Interase, Oracle, Sybase.

2.4 Arquitectura de los DBMS

El lenguaje SQL está hoy en día totalmente estandarizado, y en cambio encontramos distintas arquitecturas de DBMS. Sin embargo se pueden distinguir dos tipos generales de arquitecturas para los sistemas de bases de datos (Figura 8) y (Figura 9).

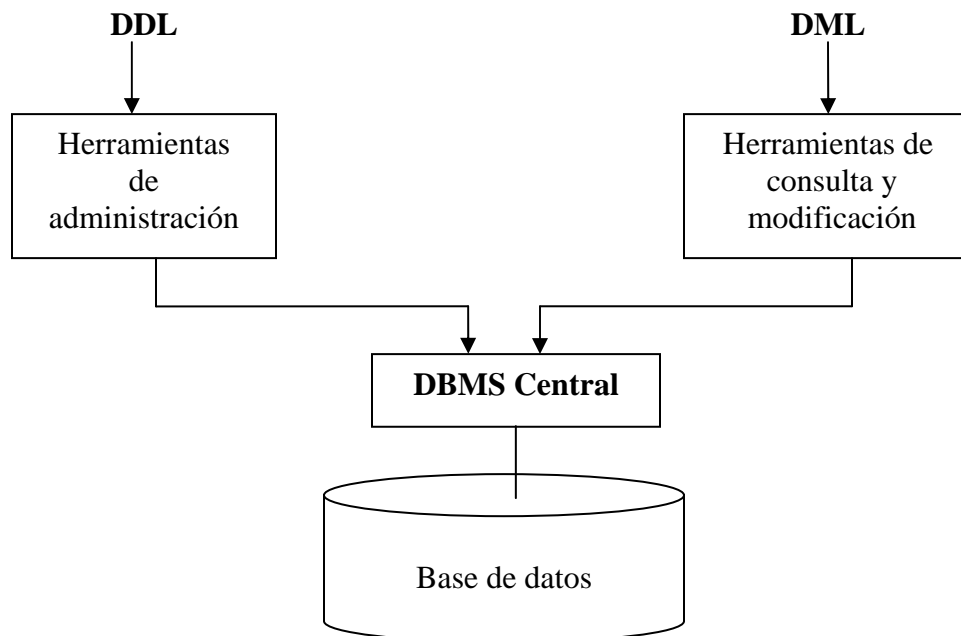


Figura 4. Arquitectura separada de DBMS

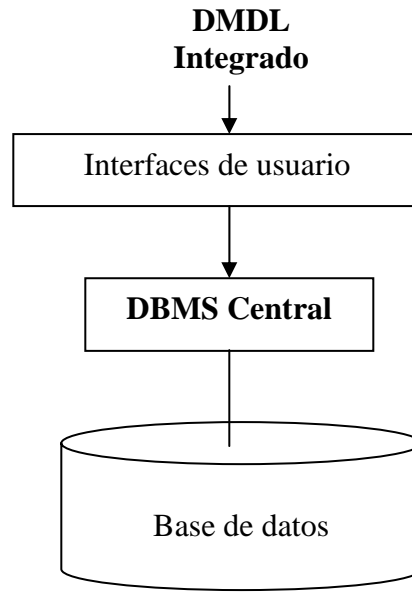


Figura 5. Arquitectura integrada de DBMS

El tipo de arquitectura integrada es en general preferible a la arquitectura separada y el más común entre los DBMS comerciales. De todos modos, la consecuencia de una integración de los lenguajes de definición de datos (DDL) y los de manipulación de datos (DML) en un sólo lenguaje (DMDL: Data Manipulation and Description Language), son a nuestro parecer positivas y negativas. Por un lado, esta integración resulta muy cómoda para el DBA, puesto que le basta con aprender un solo lenguaje formal para realizar todas las tareas de creación y mantenimiento de la base de datos. Pero por otro lado, estos sistemas (tanto los separados como los uniformes) fuerzan una proyección directa desde el nivel externo al interno, haciendo que el nivel conceptual, el fundamental según la arquitectura ANSI/X3/SPARC, desaparezca o se implemente en el nivel externo como una vista global externa. Por esta razón algunos DBA inexpertos tienden a obviar la fase de análisis, cuando de hecho es la vital para la correcta implementación de la base de datos. Un buen modelado conceptual es una condición indispensable para el correcto desarrollo de una base de datos. Lo ideal es usar un DBMS que nos permita desarrollar todas las tareas (de descripción y de manipulación) lo más fácilmente posible, pero no sin antes disponer de todas las herramientas necesarias para un correcto modelado conceptual, estén o no incluidas en el DBMS.

Arquitectura de ANSI/SPARC

Define esquemas en tres niveles (Figura 10) y realiza mapeos entre dichos niveles para transformar consultas y datos:

- **Interno:** Describe estructuras de almacenamiento físico y rutas de acceso.
- **Conceptual:** Describe la estructura y restricciones para la base de datos completa.
- **Externo:** Describe las diferentes vistas de los usuarios.

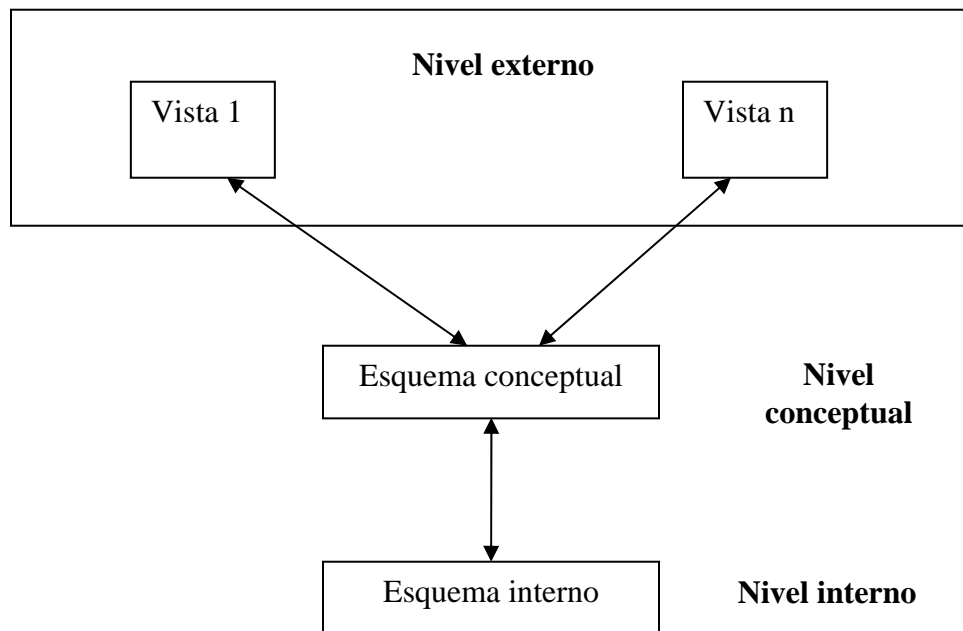


Figura 6. Arquitectura de ANSI/SPARC

2.5 Estándares ANSI-SQL

SQL³ es un lenguaje de consulta y programación de bases de datos utilizado para acceder a los datos y para consultar, actualizar y encargarse de los sistemas de bases de datos relacionales. Tanto ANSI⁴ como ISO⁵ han definido estándares para SQL. ANSI es una organización de grupos industriales y de negocios que desarrollan estándares de comunicación y negocio para los Estados Unidos. ANSI también es un miembro de ISO y de IEC⁶. ANSI publica estándares para EEUU que se corresponden con los estándares internacionales. En 1992, ISO e IEC publicaron un estándar para SQL denominado SQL-92. ANSI publicó un estándar correspondiente, ANSI SQL-92, que en EEUU se conoce algunas veces

³ SQL: Structured Query Language, Lenguaje de Consulta Estructurado

⁴ ANSI: American National Standards Institute, Instituto nacional de normalización americano

⁵ ISO: International Organization for Standardization, Organización internacional para la estandarización

⁶ IEC: International Electrotechnical Commission, Comisión electrotécnica internacional

como ANSI SQL. Aunque bases de datos relacionales diferentes utilicen versiones ligeramente diferentes de SQL, la mayoría cumple con el estándar ANSI SQL.

2.5.1 SQL-89

La historia de SQL empieza en 1974 con la definición, (por parte de Donald Chamberlin y de otras personas que trabajaban en los laboratorios de investigación IBM) de un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional. Este lenguaje se llamaba SEQUEL (Structured English Query Language) y se implementó en un prototipo llamado SEQUEL-XRM entre 1974 y 1975. Los experimentos con ese prototipo condujeron, entre 1976 y 1977, a una revisión del lenguaje (SEQUEL/2), que a partir de ese momento cambió de nombre por motivos legales, convirtiéndose en SQL.

El prototipo (System R), basado en este lenguaje, se adoptó y utilizó internamente en IBM y lo adoptaron algunos de sus clientes elegidos. Gracias al éxito de este sistema, que no estaba todavía comercializado, también otras compañías empezaron a desarrollar sus productos relacionales basados en SQL. A partir de 1981, IBM comenzó a entregar sus productos relacionales y en 1983 empezó a vender DB2. En el curso de los años ochenta, numerosas compañías (por ejemplo Oracle y Sybase, sólo por citar algunos) comercializaron productos basados en SQL, que se convierte en el estándar industrial de hecho por lo que respecta a las bases de datos relacionales.

En 1986, el ANSI adoptó SQL (sustancialmente adoptó el dialecto SQL de IBM) como estándar para los lenguajes relacionales y en 1987 se transformó en estándar ISO. Esta versión del estándar tenía el nombre de SQL/86. En 1989, ANSI definió el **SQL89**, basado en el anterior pero con una serie de mejoras (definición de claves primarias, integridad de los datos, etc). Una característica importante definida era la posibilidad de utilizarse a través de dos interfaces: interactivamente o dentro de programas de aplicación.

En su primera versión del SQL-89 se tienen tres partes:

- **El lenguaje de definición de datos (LDD).** Contiene todas las instrucciones para definir el esquema de una base de datos, como son: **create**, **alter** y **drop**.
- **El lenguaje de manipulación de datos (LMD).** Contiene las instrucciones de manejo de las tablas como son: **select**, **insert**, **delete** y **update**, y para control de concurrencia como: **commit** y **rollback**.

- **El lenguaje de control de datos (LCD).** Contiene aquellas instrucciones para dar y revocar permisos de acceso a los datos de la base de datos, como son: **grant** y **revoke**.

Todas las instrucciones pueden ir embebidas en programas escritos en otros lenguajes de programación, como: Cobol, Fortran, Pascal t PL/1.

Todas las sentencias SQL comienzan con un verbo, una palabra clave que describe lo que la sentencia hace. CREATE, INSERT, DELETE, COMMIT son verbos típicos. La sentencia continua con una o más cláusulas. Una cláusula puede especificar los datos sobre los que debe actuar la sentencia, o proporcionar más detalles acerca de lo que la sentencia debe hacer. Todas las cláusulas comienzan también con una palabra clave, tal como WHERE, FROM, INTO y HAVING. Algunas cláusulas son opcionales, otras necesarias. La estructura y contenido específico varían de una cláusula a otra. Muchas cláusulas contienen nombres de tablas o columnas; algunas pueden contener palabras claves adicionales, constantes o expresiones.

2.5.2 SQL-92

SQL-92 fue desarrollado por el comité técnico NCITS H2 sobre bases de datos. Este comité desarrolla estándares para la sintaxis y semántica de los lenguajes de bases de datos. SQL-92 fue diseñado para ser un estándar para los sistemas administradores de bases de datos relacionales (RDBMS). Esta basado en SQL-89, cuya primera versión se conoce como SQL-86. En 1992 aparece SQL2 o SQL92, la versión hoy en día más difundida ([ISO/IEC 1992] [ANSI 1992] [ISO/IEC 1994]). Con la aparición de la segunda versión del estándar (SQL2) en 1992, prácticamente todos los RDBMS, incluso los no relacionales, incluían soporte a SQL. Hoy en día, SQL se ha convertido en el lenguaje de consulta más utilizado.

SQL además de permitirnos consultas en la base de datos, contiene primitivas de definición de tablas, actualización de la base de datos, definición de vistas otorgamientos de privilegios, etc. A continuación se mostrarán aspectos del estándar ANSI de 1992, conocido como SQL-92.

Definición de Esquemas: La definición de un esquema es simple. Sólo se necesita identificar el comienzo de la definición con una instrucción CREATE SCHEMA y una cláusula adicional AUTHORIZATION y a continuación definir cada dominio, tabla, vista y demás en el esquema. El dueño del esquema, o propietario del esquema puede otorgar privilegios de acceso y actualización de la base de datos definida en el esquema a otros usuarios del sistema.

Tipos de datos y dominios: Un dominio es un conjunto del cual toma sus valores una columna de una relación. Según este concepto, los tipos de datos predefinidos son dominios. Adicionalmente SQL-92 permite la definición de dominios por parte de los usuarios

Cadenas de caracteres: Los campos de character siempre almacenan n caracteres, aún cuando tengan que rellenar con blancos a la derecha para completar la longitud n. Los campos character varying sólo almacenan el número real de caracteres que se introdujeron (hasta un máximo de n).

Cadenas de bits: Estos campos se usan para banderas u otras máscaras de bits para el control.

Fechas y horas: El tipo Date se da en el orden año, mes, día con cuatro dígitos para el año. El Time se da en horas (0 a 23), minutos, segundos y décimas de segundos. El Timestamp es la fecha más la hora.

Intervalos: Un intervalo es la diferencia entre dos fechas (año-mes) o entre dos horas (día-hora).

Definición de dominios: Los tipos de datos con restricciones (constrains) y valores por defecto (default values) se pueden combinar en la definición de dominios. Una definición de dominio es un tipo de datos especializado que puede estar definido dentro de un esquema y utilizado en la definición de columnas.

Definición de Tablas: Las tablas se definen en tres pasos:

1. Dar el nombre de la tabla.
2. Definir cada columna, posiblemente incluyendo restricciones de columna.
3. Definir las restricciones de la tabla.

2.5.3 SQL 3

El lenguaje estándar llamado SQL3, prometió ser un aumento de la segunda generación de SQL (comúnmente conocido como SQL92, debido al año de su publicación), SQL3 fue originalmente planeado para su uso en el año 1996, pero tardó 7 años en desarrollarse en vez de los tres o cuatro que se pensaba iba a tardar. Durante siete años se produjeron varios estándares que atacaban aspectos específicos de SQL estos estándares intermedios fueron:

- **CLI-95:** El estándar SQL/CLI (CLI por Call Level Interface), cuya implementación más conocida es el estándar ODBC.
- **PSM-96:** El estándar SQL/PSM (PSM por Persistent Stored Modules), el cual especifica la sintaxis de la lógica procedimental de los módulos del

servidor de SQL. A pesar de que los manejadores comerciales tenían la capacidad de usar procedimientos almacenados no había ningún estándar al respecto.

- **OLB-98:** El estándar SQL/OLB (OLB por Object Language Bindings) provee la habilidad de incluir comandos de SQL en programas de Java y está basado en el paradigma de JDBC, el cual utiliza iteradores.

SQL3 está caracterizado como “SQL orientado a objetos” y es la base de algunos sistemas de manejo de bases de datos orientadas a objetos (incluyendo ORACLE, Informix Universal Server, IBM’s DB Universal Database y Cloudscape, además de otros).

SQL:1999 envuelve características adicionales que se consideran herencia de los SQL relacionales, así como también una reestructuración de los documentos de los estándares con vista a una mayor progresión hacia normas más efectivas.

Proceso de desarrollo de normas:

Las dos organizaciones que se involucraron en la estandarización de SQL, y por lo tanto en el desarrollo de SQL: 1999 son ANSI e ISO. Más específicamente, la comunidad internacional de trabajos mediante ISO/IEC JTC1 (Joint Technical Committee 1), un comité formado por la organización internacional de estandarización junto con la comisión internacional electrotécnica. La responsabilidad de las JTC1 es desarrollar y mantener la información relativa a la tecnología. Dentro de JTC1, el subcomité SC32, cuya función era el intercambio y gestión de datos se formó para la estandarización de normas relativas a varias bases de datos y metadatos que habían sido desarrollados por otras organizaciones (tal como el ahora disuelto SC21). SC32, es a la vez, un número de grupos de trabajo que actualmente realizan los trabajos técnicos-WG3(lenguajes de bases de datos) es responsable de las normas de SQL, mientras WG4 está desarrollando el SQL/MM (SQL multimedia, un departamento de normas que especifiquen las bibliotecas de tipos usando facilidades de SQL orientado a objetos).

En los Estados Unidos, IT estándares son manejados por la institución de acreditación y desarrollo de normas nacionales estadounidenses, el comité NCITS (Comité nacional para la estandarización de tecnología de la información, anteriormente conocido como X3). NCITS comité técnico H2 (anteriormente X3H2) es responsable de varios estándares relativos a la gestión de datos, incluyendo SQL y SQL/MM.

Cuando la primera generación de SQL fue desarrollada (SQL-86 y su aumento menor SQL-89), casi todo el proceso se hizo en Estados Unidos por X3H2 y otras naciones participaron en su mayor parte en el modo de revisar y criticar el trabajo propuesto por ANSI.

En el momento que SQL-89 fue publicado, la comunidad internacional hizo por escrito propuestas para la especificación que al final llegó a ser SQL-92; que no ha cambiado mientras SQL:1999 está siendo desarrollado.

Las primeras versiones de la norma son conocidas como SQL-86 (o SQL-87, porque la versión no fue publicada hasta 1987), SQL-89 y SQL-92, mientras que la versión actual debe llegar a ser conocida como SQL: 1999. ¿Por qué no SQL-99?, Simplemente porque se debe pensar en el nombre de la próxima generación y SQL-02 puede ser confundido con SQL2 (que era el proyecto bajo el cual fue desarrollado SQL-92). En otras palabras, se espera que desde el año 2000 no se produzcan problemas con los nombres de SQL.

Contenidos de SQL:1999

A continuación se van a describir los aspectos nuevos de esta generación en desarrollo de SQL. Los aspectos pueden ser divididos en “aspectos relacionales” y “aspectos relacionados con objetos”.

Aspectos relacionales: Es más adecuado llamar a esta categoría como “aspectos que relacionan el papel de SQL en el modelado de datos”. Estos aspectos no están estrictamente limitados al modelo relacional, pero no están relacionados con la orientación a objetos.

Estos aspectos se dividen en cinco grupos: nuevos tipos de datos, nuevos predicados, semántica mejorada, seguridad adicional, la base de datos activa. Hablaremos de cada uno de los grupos por separado.

Nuevos tipos de datos: SQL:1999 tiene cuatro nuevos tipos de datos (aunque alguno de ellos tiene variantes identificables). El primero de estos tipos es LARGE OBJECT(objeto grande) o LOB. Otro tipo de dato nuevo es el BOOLEAN, que permite a SQL registrar valores de verdad, falso y desconocido. Complejas combinaciones de predicados pueden ser ahora expresadas de una manera más sencilla que antes. SQL:1999 también tiene dos nuevos tipos compuestos: ARRAY y ROW. El tipo ARRAY permite almacenar una colección de valores directamente en una columna de una tabla.

Nuevos predicados: SQL:1999 tiene tres nuevos predicados, uno de los cuales se analizará conjuntamente con la orientación a objetos. Los otros dos son el predicado SIMILAR y el predicado DISTINTO. Desde la primera versión del SQL estándar, las cadenas de caracteres están limitadas a simples comparaciones (como =, > ó <>) y las rudimentarias capacidades del predicado like.

Nueva semántica en SQL: 1999: Es difícil saber exactamente fijar un límite a la hora de hablar de la nueva semántica en SQL: 1999, pero se dará una selección de lo que se cree más importante. Una de las demandas de los escritores de

aplicaciones era ampliar las clases de vistas. Muchos entornos usan vistas pesadas como mecanismo de seguridad y/o como simplificación de una aplicación vista de la base de datos. Como siempre, si la mayoría de vistas no son actualizables, luego estas aplicaciones frecuentemente tienen que escapar de los mecanismos de vista y confiar directamente en la modificación de las tablas subyacentes; esto es una situación poco satisfactoria. SQL:1999 ha incrementado significativamente el número de vistas que pueden ser modificadas directamente, usando solo lo que se facilitó en el estándar. Esto depende en gran medida de las dependencias funcionales para ir determinando que vistas pueden ser modificadas, y como hacer estos cambios en la tabla de la base de datos para efectuar estas modificaciones. Otra deficiencia criticada en gran medida de SQL era la imposibilidad de construir aplicaciones recursivas. SQL:1999 está provisto para hacer llamadas recursivas, cosa que satisface lo dicho con anterioridad. Escribiremos la pregunta en base a la cual deseamos hacer la recursión y daremos un nombre. Luego usaremos el nombre asociando la pregunta a una expresión. Se ha mencionado a los locators como un valor del cliente que puede representar un valor LOB almacenado en el servidor. Los locator pueden ser usados en algunos casos para representar los valores de un ARRAY, aceptando el hecho de que (como LOB) los array pueden ser frecuentemente demasiado grandes para ser pasado convenientemente entre una aplicación y la base de datos. Los locator pueden ser usados también para representar tipos de valores de datos indefinidos que también tienen la potencialidad de ser grandes y abultados. Finalmente, SQL:1999 ha agregado la noción de savepoints, ampliamente implementado en sus productos. Un savepoint es como una pequeña substracción en la que una aplicación puede deshacer las acciones realizadas después de comenzar el savepoint sin deshace todas las acciones de una transacción entera.

Mejoras de Seguridad: Las nuevas facilidades de seguridad en SQL:1999 tienen un papel muy importante. Los privilegios pueden ser otorgados según un rol y este a su vez puede otorgar privilegios individuales para otros roles. Esta estructura anidada mejora el manejo de la seguridad en el ambiente de una base de datos. Los roles han estado siendo implementados por los productos SQL desde hace varios años (aunque ocasionalmente bajo nombres diferentes).

Base de datos activa: SQL:1999 reconoce la noción de base de datos activa. Esto es facilitado por lo que se conoce como triggers (disparadores). Un trigger, como se sabe, es una facilidad que permite a los diseñadores de bases de datos realizar operaciones seguras siempre que una aplicación realice operaciones en tablas particulares.

Orientación a objetos: Además de las características discutidas hasta ahora, SQL:1999 se caracteriza porque fue desarrollado principalmente para manejar objetos. Algunas de las características que están dentro de esta categoría fueron definidas en el estándar SQL/PSM publicado en 1996 específicamente para llamadas a funciones y procedimientos desde SQL. SQL: 1999 mejora esta

capacidad que llamó SQL-invoked routines, para añadir una tercera clase de rutina conocida como método, que luego veremos.

2.6 Tareas y funciones del administrador de la base de datos (DBA)

El DBA debe administrar la actividad y estructura de la base de datos, además de tener el control centralizado sobre el DBMS.

El DBA es responsable primordialmente de:

- Administrar la estructura de la Base de Datos
- Administrar la actividad de los datos
- Administrar el Sistema Manejador de Base de Datos
- Establecer el Diccionario de Datos
- Asegurar la confiabilidad de la Base de Datos
- Confirmar la seguridad de la Base de Datos.

La administración de la estructura de la base de datos requiere que el DBA participe en el diseño inicial de la misma y su puesta en práctica así como controlar y administrar sus requerimientos, ayudando a evaluar alternativas, incluyendo los DBMS a utilizar. El DBA participa en el desarrollo de procedimientos y controles para asegurar la calidad y la alta integridad de la base de datos.

Debido a que en el medio de la base de datos están involucrados muchos usuarios, es necesario establecer una función administrativa permanente para coordinar y llevar a cabo todos los pasos de diseño, implantación y mantenimiento de una base de datos integrada. El objetivo de esta función administrativa es analizar las diferentes necesidades y resolver conflictos de intereses, especialmente cuando se deben incluir procedimientos y políticas mediante las cuales los usuarios puedan registrar sus necesidades de modificaciones, así la comunidad podrá analizar y discutir los impactos de dichas modificaciones, determinándose entonces la puesta en práctica de tales alteraciones.

En función del tamaño y complejidad de una base de datos y de sus aplicaciones, las modificaciones pudieran tener resultados inesperados. El DBA debe estar preparado para reparar la base de datos y reunir suficiente información para diagnosticar y corregir el problema provocado por la falla. Después de un cambio la base de datos es más vulnerable a fallas. Por lo anterior es necesario

que el DBA sea responsable de la documentación de las modificaciones que han sido efectuadas, como fueron realizadas y cuando fueron establecidas, puesto que una modificación sobre la estructura de la base de datos puede ocasionar un error a corto plazo. Al no tener la documentación sobre las modificaciones realizadas, el diagnóstico resultaría complicado y sería necesaria una secuencia de reejecuciones para intentar detectar el punto en conflicto, el riesgo de este procedimiento radica en que es posible afectar la información contenida en la base de datos.

Una razón más para documentar consiste en la necesidad de mantener organizados datos históricos. Ocurre comúnmente que se desea realizar una consulta sobre los respaldos para conocer el estado que guardaba la información en un periodo determinado que transcurrió previamente. Los registros de modificación existentes en la documentación permitirán resolver problemas de incompatibilidad entre las estructuras que eran vigentes en el periodo de respaldo y las que lo son ahora; permitirá también el desarrollo de módulos de ajuste que faciliten la traducción de formatos y/o escalas para valores almacenados. En los casos de caídas del sistema se presenta una situación parecida; los respaldos son requeridos y habrá de verificarse su estructura, formato y escala para integrarlos a la operación del sistema.

La tarea de la documentación es cada vez más ligera y precisa cuando se utilizan DBMS que integran herramientas CASE para las tareas de diseño, mantenimiento y documentación. Estas mismas herramientas CASE proporcionan, en la mayoría de los casos, la facilidad de generar y mantener en forma automática el Diccionario de Datos.

La administración de la base de datos es así un cargo compuesto por gente responsable de proteger un valioso recurso: los datos. En el medio convencional del procesamiento de datos, un programador de aplicaciones “posee” un archivo de datos, mientras que los usuarios “protegen” sus datos para evitar que otros los usen. La era de la base de datos ha eliminado la idea de “propiedad” individual. El administrador de la base de datos no es el “propietario” de los datos, sino el “protector” de ellos. Ya que al programador de aplicaciones se le “quita” el control directo sobre los datos, pierden el sentimiento de contacto personal y responsabilidad de éstos. Esta falta de contacto obliga a las empresas a desarrollar procedimientos para asegurar que la integridad de los datos no quede comprometida. Este objetivo deberá estar coordinado con la función de administración de la base de datos.

La administración de la base de datos es una función que proporciona servicios a los usuarios de la base de datos. El DBA protege al recurso llamado “datos”. La función del DBA deberá ser situada lo bastante alto en la jerarquía de la organización para tener autoridad, así como responsabilidad, sobre las estructuras de datos y su acceso. La persona a cargo de la función del DBA también debe conocer la forma en que trabaja la empresa y cómo usa los datos.

Aunque el DBA proteja los datos, no los procesa. El DBA no es usuario del sistema, en consecuencia, no administra valores de datos, el DBA administra actividad de datos. Dado que la base de datos es un recurso compartido, el DBA debe proporcionar estándares, guías de acción, procedimientos de control y la documentación necesaria para garantizar que los usuarios trabajen en forma cooperativa y complementaria al procesar datos.

La concurrencia de múltiples usuarios requieren de estandarizar los procesos de operación, el DBA es responsable de tales especificaciones y de asegurarse que estas lleguen a quienes concierne. Todo el ámbito de la base de datos se rige por estándares, desde la forma como se capture la información (tipo, longitud, formato), como es procesada y presentada. Entre las funciones del DBA se encuentra la de revisarlos periódicamente para determinar su operatividad, y en su caso ajustarlos, ampliarlos o cancelarlos. Es también su responsabilidad el que estos se cumplan. Cuando se definen estándares sobre la estructura de la BD, estos deben registrarse en una sección del diccionario de datos a la que todos aquellos usuarios relacionados con ese tipo de proceso pueden acceder.

Otro de los aspectos que el administrador debe atender es el de coordinar las nuevas propuestas para realizar ajustes en los derechos de acceso a datos compartidos y aplicaciones para determinar si pudieran aparecer problemas cuando dos o más grupos de usuarios quedan autorizados para modificar los mismos datos. Uno de tales conflictos es el de la actualización perdida, que ocurre cuando el trabajo de un usuario queda sobrescrito sobre el de un segundo usuario. El DBA queda responsabilizado para identificar la posible ocurrencia de dichos problemas, así como de crear normas y procedimientos para su eliminación.

Entre las alternativas mas utilizadas por el DBA para tratar de resolver o minimizar este problema se encuentran las siguientes:

- a) Restringir el acceso a los procedimientos para ciertos usuarios.
- b) Restringir al acceso a los datos para ciertos usuarios procedimientos y/o datos.
- c) Evitar la coincidencia de horarios para usuarios que comparten.

Las técnicas de recuperación son otra función esencial del DBA al administrar la actividad de datos. A pesar de que el DBMS lleva a cabo una parte del proceso de recuperación, los usuarios determinan en forma crítica la operatividad de esos sistemas de protección. El DBA debe anticipar fallas y definir procedimientos estándares de operación, los usuarios deben saber que hacer cuando el sistema este caído y que es lo primero que debe realizarse cuando el sistema este puesto en marcha nuevamente. El personal de operación deberá saber como iniciar el proceso de recuperación de la base de datos, que copias de seguridad utilizar; como programar la reejecución del tiempo perdido y de las tareas pendientes, es importante también establecer un calendario para llevar a

cabo estas actividades sin afectar a otros sistemas dentro de la organización que hagan uso de los mismos recursos de computo.

A demás de administrar la actividad de datos y la estructura de la base de datos, el DBA debe administrar el DBMS:

- Debe compilar y analizar estadísticas relativas al rendimiento del sistema e identificar áreas potenciales del problema: requiere investigar todas las quejas sobre el tiempo de respuesta del sistema, la precisión de los datos y la facilidad de uso.
- Debe vigilar periódica y continuamente las actividades de los usuarios en la base de datos: analizará estos datos para determinar si se necesita una modificación en el diseño de la base de datos para manejar su rendimiento o para facilitar las tareas de los usuarios.
- Debe analizar las estadísticas de tiempo de ejecución sobre la actividad de la base de datos y su rendimiento: cuando se identifique un problema de rendimiento, deberá determinar si resulta apropiada una modificación a la estructura de la base de datos o al sistema.
- Debe administrar y controlar la migración tanto de las estructuras, como de los datos y las aplicaciones: cuando el fabricante del DBMS en uso anuncie una nueva versión del producto, debe realizarse un análisis de las características que esta incorpora y confrontarlas contra las necesidades de la comunidad de usuarios. Si se decide la adquisición del producto, los usuarios deben ser notificados y capacitados en su uso. El software de soporte y otras características de hardware pueden implicar también modificaciones de las que el DBA es responsable ocasionalmente, estas modificaciones traen como consecuencia cambios en la configuración o en algunos parámetros de operación del DBMS.



3 ANÁLISIS DE HERRAMIENTAS USADAS PARA EL DESARROLLO DEL SISTEMA

En este capítulo se analizan y se describen las herramientas Web, que utilizamos para nuestro proyecto.

Análisis de herramientas usadas para el desarrollo del sistema

Hoy en día diseñar un sistema que maneje bases de datos mediante aplicaciones Web se ha convertido en una labor compleja puesto que es necesario administrar y gestionar correctamente las bases de datos. La Web es aquí entendida como interfaz de software que permite una serie de funcionalidades como que el usuario pueda interrogar y consultar de forma directa a la base de datos y obtener las referencias o el acceso directo a los recursos o documentos buscados (figura 11).

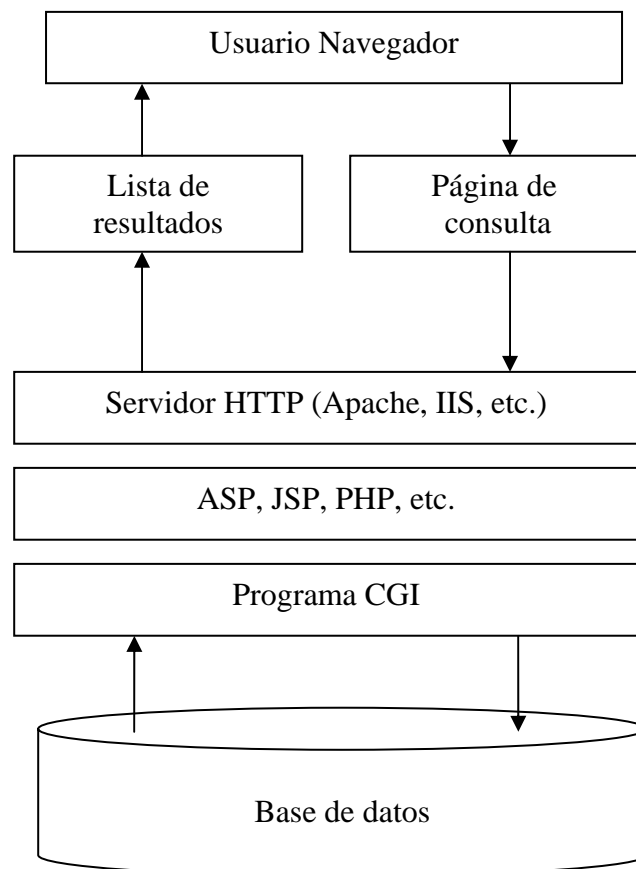


Figura 1. Interactividad de usuario con bases de datos vía Web

Los DBMS suelen incluir herramientas de administración que permiten ajustar el rendimiento en función de las necesidades particulares.

Existen sistemas de administración de bases de datos tanto de uso libre, como soluciones comerciales de pago. Una de las tendencias más claras en la Web actual es integrar el acceso a datos en los servidores de aplicaciones y esto ha conducido a que casi todos los fabricantes de sistemas de administración de

bases de datos comerciales ofrezcan sus propios servidores de aplicaciones que se integran a bajo nivel con los productos de bases de datos de la misma empresa. Como ejemplos, tenemos Sybase Enterprise Server y Oracle Application Server.

Un servidor de aplicaciones no es más que un cambio de nombre para algunos servidores Web de nueva generación que permiten construir aplicaciones. Suelen asociarse con servidores de alto rendimiento pensados para dar servicio a sitios Web con grandes necesidades para gestionar movimientos de datos, afluencia de visitas, atención de transacciones hacia bases de datos, etc. Generalmente los fabricantes del sector tienen a disposición del público un servidor Web básico y otro con multitud de extensiones integradas al que llaman servidor de aplicaciones.

3.1 Sistemas Administradores de Base de Datos Relacionales (RDBMS)

Se conoce como RDBMS al software que administra bases de datos usando el modelo entidad-relación. Existe una gran variedad de RDBMS, a continuación se presenta una tabla con las características principales de algunos RDBMS para poder determinar cuál es el más conveniente para el desarrollo de nuestro trabajo de tesis.

Producto	Vendedor	Libre o comercial	Tipo de manejador	Descripción	Requerimientos
Oracle	Corporación Oracle	Ambos	(RDBMS) Versión 7.x. (ORDBMS) Versión 8.x (OODBMS) Versión 9.x	Este manejador se ocupa de garantizar el aprovechamiento del sistema al máximo en ambientes cargados de información. Buena capacidad de almacenar un conjunto de datos y acude a estos de forma consecuyente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas. Es el mayor y más usado Sistema Manejador de Bases de Datos Relacionales (RDBMS) en el mundo. La	Procesador: Pentium 166 MHz o superior, mínimo 128 Mb RAM (256 Mb RAM si se usan utilidades JAVA), 1 Gb Disco Duro. Entre 298 y 587 Mb disponible de espacio en disco. Sistema Operativo: Windows NT 4.0, Windows 2000, Windows98 o Windows95 Servidor: Mínimo 64 Mb RAM, de los cuales 32 Mb estarán disponibles para el System Global Area (SGA) Para cada Repositorio: Aproximadamente 140 Mb en el SYSTEM tablespace para los paquetes del repositorio,

				<p>Corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Además incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios. Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta 17 idiomas, porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.</p>	<p>procedimientos y vistas entre 20 y 325 Mb en otras tablespaces, para los datos del repositorio Oracle8i Enterprise Edition ó Standard Edition, release 8.1.7* versión de SQL*Plus compatible con su Base de datos versión de TNS Listener compatible su Base de Datos</p> <p>Plataformas soportables: Hace uso de los recursos del sistema informático en todas las arquitecturas de hardware. Por lo tanto, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Por ejemplo:</p> <ul style="list-style-type: none"> • Linux x86 • z/Linux • Linux Itanium • Microsoft Windows (32-bit) • Microsoft Windows (64-bit Itanium) • Solaris x86 • Solaris SPARC (64-bit) • AIX5L • HP-UX PA-RISC • HP-UX Itanium • HP Tru64 UNIX • HP OpenVMS Alpha • IBM z/OS • Mac OS X Server <p>Requerimientos de hardware:</p> <ul style="list-style-type: none"> • 48 MB de RAM para correr la BD • 200 MB adicionales para el Enterprise Manager
--	--	--	--	---	---

<p>SQL server</p>	<p>Microsoft</p>	<p>Comercial</p>	<p>(RDBMS) SQL Server 2000</p>	<p>Se puede describir como la mejor base de datos para Windows NT. SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.</p> <p>La estrategia de Microsoft es la de hacer que SQL Server sea la base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.</p>	<p>Existen tres tipos de instalación de SQL SERVER 7.0 para lo cual se utilizan plataformas de Microsoft, a continuación se detalla cada requerimiento:</p> <ul style="list-style-type: none"> • Standard Edition: Microsoft Windows NT server 4.0 o superior • Enterprise Edition: Microsoft Windows NT advanced Server 4.0 o superior • Desktop Edition: Microsoft Windows NT, Microsoft Windows NT Workstation, Windows 95, Windows 98 <p>Plataformas soportables:</p> <ul style="list-style-type: none"> • Microsoft Windows (32-bit) • Microsoft Windows (64-bit Itanium) <p>Requerimientos de hardware: Computadora: DEC Alpha AXP y sistemas compatibles, Intel o compatible (Pentium 166 MHz o superior, Pentium PRO, o Pentium II) Memoria: 32 MB de RAM Unidades de disco: Unidad de CD-ROM, más un disco duro con al menos 80 MB de espacio disponible en disco para la instalación mínima. Requerimientos mínimos de espacio:</p> <ul style="list-style-type: none"> • Completa: 210 MB • Típica: 185 MB • Herramientas de Administración: 90 MB • Mínima: 80 MB
--------------------------	------------------	------------------	--	---	---

<p>Informix Dynamic Server</p>	<p>IBM</p>	<p>Comercial</p>	<p>(RDBMS) Dynamic Server (OREDBMS) Universal Server</p>	<p>IBM Informix Dynamic Server (IDS) proporciona fiabilidad superior, atendiendo las necesidades de las exigentes prácticas actuales del comercio virtual particularmente para aplicativos que requieran transacciones de alto desempeño. Soporta requisitos de procesamiento de transacción online, complejos y rigurosos. Optimiza capacidades de inteligencia del negocio competitivas. Maximiza operaciones de datos para el grupo de trabajo y para la empresa en total. Proporciona la firmeza de una administración de base de datos comprobada, mejor de su especie.</p>	<p>Los requerimientos de Informix varían de acuerdo a los paquetes que piense adicionar al programa, he aquí un claro cuadro de los requerimientos. Requerimientos típicos de RAM Small multi user (1-10 usuarios): 4MB-8MB Large multi user: 16MB+ Uso típico de memoria: Programmer's environment: 2.67 MB + db engine P-code runner: 2.75 MB + db engine Database engine: Informix dynamic server para Unix: 22MB Informix-SE para Unix: 6MB Disk usage: 68 MB este es el tamaño del archive comprimido Run time: 20 MB este es el tamaño del archivo comprimido. Plataformas soportables: <ul style="list-style-type: none"> • Linux • UNIX • Windows NT • MSWindows • Xwindows Requerimientos de hardware: Memoria RAM entre 4 y 22 MB</p>
<p>Sybase</p>	<p>Sybase Inc.</p>	<p>Comercial</p>	<p>(RDBMS) System 10/11</p>	<p>Sybase ASE se ha establecido como una de las plataformas de administración de datos más económicas para la computación crítica. Sybase tiene una infraestructura abierta y basada en estándares que soporta:</p> <ul style="list-style-type: none"> • Plataformas J2EE, JSP, MVC, 	<p>Los requerimientos varían de acuerdo a las necesidades del Cliente, tomando en cuenta parámetros como:</p> <ul style="list-style-type: none"> • Volumen de la información • Numero de usuarios que utiliza • Plataforma que utiliza (Windows NT y UNIX)

				<p>y XML</p> <ul style="list-style-type: none"> Arquitectura stateless, distribuida y escalable Generación asincrónica de portlets Arquitectura abierta Estándares abiertos Servidores de aplicación Tomcat, Web Logic, Web Sphere, y Sybase EAServer Bases de datos Oracle, Sybase ASE, y Sybase ASA Despliegue global 	<ul style="list-style-type: none"> Aplicaciones a ser ejecutadas <p>Plataformas soportables:</p> <ul style="list-style-type: none"> Linux x86 z/Linux Linux Itanium Unix Microsoft Windows (32-bit) Microsoft Windows (64-bit Itanium) <p>Requerimientos de hardware: En esencia, depende de qué se quiere hacer con la base de datos (que tan larga, cuantos usuarios, etc)</p>
MySQL	MySQL AB	Ambos	RDBMS y ofrece motor de base de datos orientada a objetos	<p>MySQL opera en un ambiente de red usando la arquitectura cliente / servidor. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de</p>	<p>Para Windows:</p> <ul style="list-style-type: none"> Pentium 166MHz o más Microsoft Windows 95/98/Me/NT4/2000 /XP 32 MB RAM (64 MB o más) 10MB libres en HD para el programa de instalación Windows tarjeta de video con resolución de al menos 800x600 (1024x768 es recomendada) Windows mouse compatible Posibilidad para conectarse a MySQL server <p>Para Linux:</p> <ul style="list-style-type: none"> Pentium 166MHz o más recomendado Linux OS con librería glibc versión 2.1 o más 32 MB RAM (64 MB o más)

				transacciones o con acceso a disco no transaccional ultrarrápido.	<ul style="list-style-type: none"> • 10MB de espacio en HD para el programa de instalación • XWindow tarjeta de video con al menos 800x600 de resolución (1024x768 es recomendada) • XWindow mouse compatible • Posibilidad para conectarse a MySQL server <p>Plataformas soportables:</p> <ul style="list-style-type: none"> • Linux x86 • Microsoft Windows (32-bit) • Mac OS X Server <p>Requerimientos de hardware: MySQL puede ser utilizado en equipos con hardware limitado y sin requerimientos especiales. Tampoco requiere mucha memoria RAM.</p>
PostgreSQL	PostgreSQL (Universidad de California)	Libre	Manejador de base de datos objeto-relacional	PostgreSQL es un RDBMS orientado a objetos que ha sido catalogado como el manejador de bases de datos de código abierto más poderoso del mundo. Este manejador soporta el estándar SQL en casi todas sus agrega además la posibilidad de definición de datos, herencia y funciones. Todo ello lo convierte en un manejador versátil, confiable y poderoso que ha demostrado su capacidad en el manejo de bases de datos enormes con miles de millones de	<p>Para Windows:</p> <ul style="list-style-type: none"> • Pentium 166MHz o más • Microsoft Windows 95/98/Me/NT4/2000 /XP • 32 MB RAM (64 MB o más) • 10MB libres en HD para el programa de instalación • Windows tarjeta de video con resolución de al menos 800x600 (1024x768 es recomendada) • Windows mouse compatible • Posibilidad para conectarse a PostgreSQL server

				registros y sistemas distribuidos alrededor del mundo con el soporte de la filosofía de código abierto.	<p>Para Linux:</p> <ul style="list-style-type: none"> • Pentium 166MHz o más recomendado • Linux OS con librería glibc version 2.1 o más • 32 MB RAM (64 MB o más) • 10MB de espacio en HD para el programa de instalacion • XWindow tarjeta de video con al menos 800x600 de resolución (1024x768 es recomendada) • XWindow mouse compatible • Posibilidad para conectarse a PostgreSQL server <p>Plataformas soportables:</p> <ul style="list-style-type: none"> • Linux • UNIX AIX • BSD • HP-UX • SGI IRIX • Mac OS X • Solaris • SunOS, • Tru64 • BeOS • Windows <p>Requerimientos de hardware: Postgresql puede ser utilizado en equipos con hardware limitado y sin requerimientos especiales. Tampoco requiere mucha memoria RAM.</p>
--	--	--	--	---	--

Tabla 1. RDBMS más populares

Pese a la gran variedad de RDBMS existente en el mercado, debemos considerar si adquirir el software está dentro de nuestras posibilidades económicas, de lo contrario es necesario considerar una versión libre, después verificar si se puede instalar en nuestro sistema operativo y si nuestro servidor o computadora tiene los requerimientos necesarios para que el RDBMS pueda funcionar sin ninguna dificultad.

3.2 Servidor Web

Un servidor Web es un programa que se ejecuta en un servidor (máquina que almacena y administra los sitios Web) con el propósito de atender y responder a las diferentes peticiones de los navegadores de los usuarios, proporcionando los recursos que soliciten usando el protocolo HTTP ¹ o el protocolo HTTPS ² (versión cifrada y autenticada). Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

Un servidor Web básico cuenta con un esquema de funcionamiento muy simple basado en ejecutar infinitamente el siguiente ciclo:

1. Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió petición.
5. Vuelve al segundo punto.

Un servidor Web que siga el esquema anterior cumplirá todos los requisitos básicos de los servidores HTTP, aunque sólo podrá mostrar archivos estáticos. A partir del anterior esquema se han diseñado y desarrollado todos los servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGIs, Servlets, etc.) que pueden atender peticiones, en función de que sean o no sean multi-proceso o multi-hilo, etc.

Todos los servidores Web deben incluir, al menos, la capacidad para mostrar los archivos estáticos que se encuentren en alguna parte del disco. Un requisito básico es la capacidad de especificar qué parte del disco se servirá. No resulta recomendable que el programa servidor obligue a usar un directorio concreto, aunque sí puede tener uno por defecto. Algunos servidores Web permiten también especificar directivas de seguridad (quién puede acceder a los recursos), mientras que otros hacen posible la especificación de los archivos que se deben considerar como índice del directorio.

Uno de los aspectos fundamentales del servidor Web que se elija es el nivel de soporte que ofrece para mostrar contenido dinámico. La mayor parte de los servidores Web ofrecen soporte para CGI (se debe recordar que los CGI son el

¹ HTTP: Hyper Text Transfer Protocol, Protocolo de transferencia de hipertexto.

² HTTPS: Secure Hyper Text Transfer Protocol, Protocolo Seguro de Transferencia de Hipertexto

método más antiguo y sencillo para generar contenido dinámico). Otros muchos ofrecen soporte para algunos lenguajes de programación (normalmente lenguajes interpretados) como PHP, JSP, ASP, etc. Es muy recomendable que el servidor Web que vayamos a utilizar proporcione soporte para algunos de estos lenguajes, especialmente PHP, sin tener en cuenta JSP, que normalmente requerirá un software externo para funcionar (como un contenedor de Servlets). La oferta es muy amplia, pero antes de elegir un lenguaje de programación de servidor se debe plantear si se desea un lenguaje muy estándar para que la aplicación no dependa de un servidor Web o una arquitectura concreta o si, al contrario, la portabilidad no es prioritaria y sí lo es alguna otra prestación concreta que pueda ofrecer algún lenguaje de programación concreto.

Aplicaciones Multinivel

Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: interfaz de presentación, lógica de la aplicación y los datos. La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación. La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola (Figura 12).

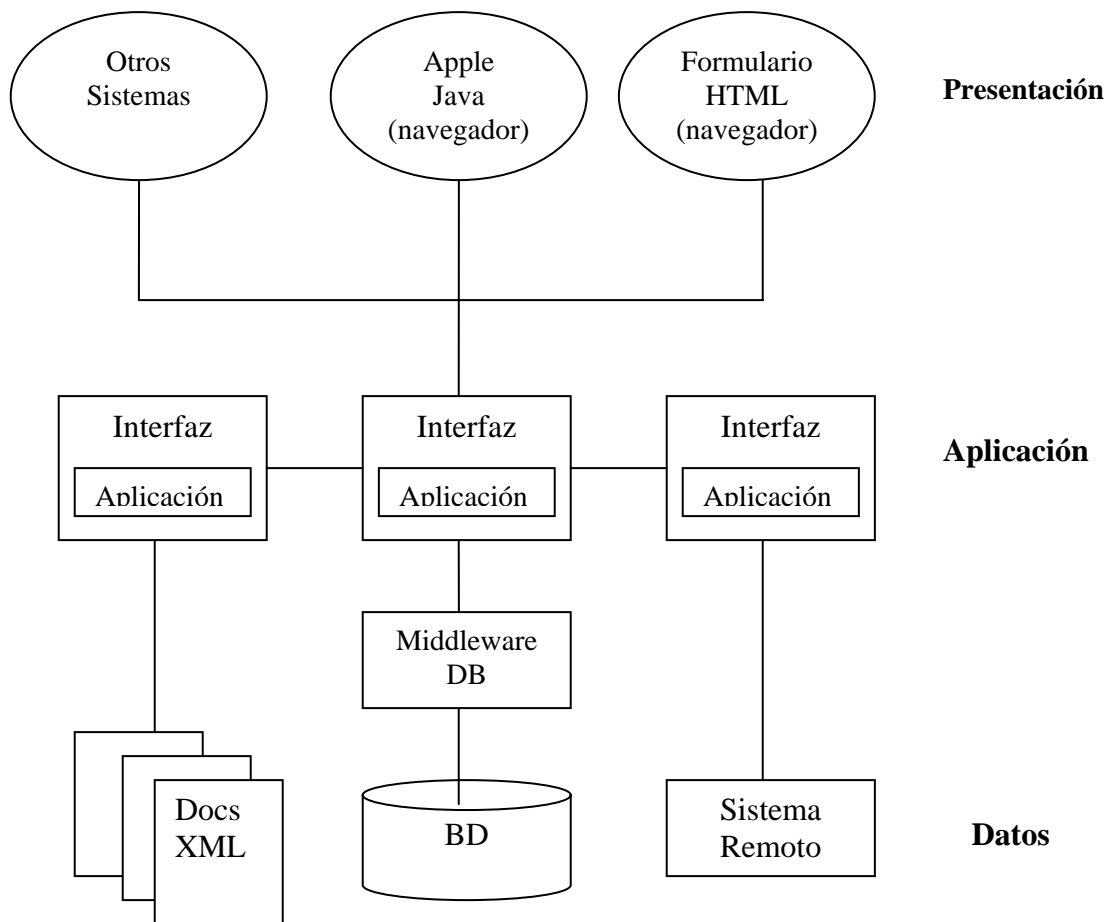


Figura 2. Arquitectura de aplicaciones Web representada en esquema multinivel

La arquitectura de las aplicaciones Web suele presentar un esquema de tres niveles (Figura 13). El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

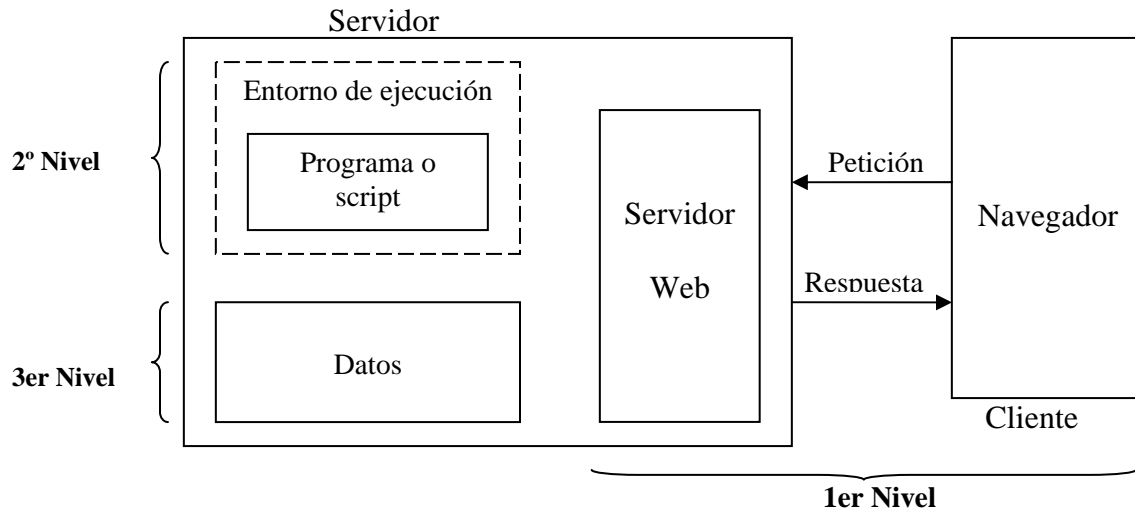


Figura 3. Arquitectura de aplicaciones Web representada en esquema de tres niveles

A continuación se muestra una lista de los servidores Web existentes

Nombre del Servidor	Sistema operativo
4D WebSTAR	
AOLserver	Unix Windows 2000 Windows 95/98 Windows ME Windows NT 4.0 Windows Server 2003 Windows XP
Apache	NetWare OS/2 Unix Linux Windows 95/98 Windows NT 4.0
BadBlu	Windows 2000 Windows 95/98 Windows ME Windows NT 4.0 Windows XP
Baikonur Web App Server	Windows 95/98 Windows NT 4.0
Covalent Enterprise Ready Server	AIX HP-UX Linux Red Hat Linux

	SUSE Linux Solaris Windows 2000 Windows Server 2003
ESAWEB	VM/CMS
Enterprise WebServer for NetWare	NetWare
GoAhead WebServer	Linux NetWare Solaris Windows 2000 Windows 95/98
Hawkeye	Linux
iTools	MacOS X Server
Java Server	HP-UX IRIX Linux OS/2 Solaris Windows 95/98 Windows NT 4.0
Jigsaw	Java_VM Solaris Windows 95/98 Windows NT 4.0
Microsoft Internet Information Services	Windows Server 2003
RapidSite	BSD IRIX Windows 2000 Windows Server 2003
RomPager Embedded Web Server	Embedded
Roxen WebServer	Linux MacOS X Server Solaris Windows 2000 Windows Server 2003 Windows XP
Sambar Server	Red Hat Linux Windows 2000 Windows 95/98 Windows ME Windows NT 4.0 Windows Server 2003 Windows XP
Savant	Windows 2000

	Windows 95/98 Windows ME Windows NT 4.0
Servotec Internet Serve	AIX HP-UX Linux Solaris Windows 2000 Windows 95/98 Windows ME Windows NT 4.0 Windows Server 2003 Windows XP
Shadow Web Server	MVS
SimpleServer:WWW	Windows 2000 Windows 95/98 Windows ME Windows NT 4.0 Windows XP
Sun Java System Web Server	Red Hat Linux Solaris Windows 2000 Windows XP
Tcl Web Server	Linux MacOS Unix Windows NT 4.0
URL Live!	Windows 2000 Windows 95/98 Windows ME Windows NT 4.0 Windows XP
Viking	Windows 2000 Windows 95/98 Windows NT 4.0 Windows XP
VqServer	AIX BSD Be OS Digital UNIX HP-UX IRIX Java_VM Linux MacOS OS/2 SCO OpenServer Solaris Windows 95/98 Windows NT 4.0

WN	AIX BSD Digital UNIX HP-UX IRIX Linux SCO OpenServer Solaris
WebBase	Windows 2000 Windows 95/98 Windows NT 4.0
Xitami	AIX BSD Be OS Digital UNIX HP-UX Linux OS/2 SCO OpenServer Solaris VMS Windows 2000 Windows 95/98 Windows ME Windows NT 4.0 Windows Server 2003 Windows XP
Zeus Web Server	AIX BSD HP-UX IRIX Linux MacOS X Server SCO OpenServer Solaris Tru64 Unix

Tabla 2. Servidores Web existentes

Como podemos ver en la lista anterior existen muchos servidores Web, los más populares son IIS, Apache y Tomcat. La intención de presentar este listado es para que el usuario pueda apreciar el gran número de servidores Web que existen y que sepa en que sistemas operativos pueden ser instalados.

3.3 Herramientas de programación

Dada la importancia y popularidad que adquiriría la Web con el paso del tiempo por encima de otros medios de comunicación, su interacción con los

usuarios también aumentaba. Debido a la constante necesidad de desarrollar páginas web que no solo se limitaran a mostrar información, fue necesario buscar alternativas a las tecnologías del lado del cliente (como HTML y Javascript). Así fue como nacieron tecnologías como PHP, ASP, JSP o ColdFusion. A continuación se explicaran las características esenciales de dichas tecnologías.

3.3.1 Tecnología PHP

PHP es un lenguaje de programación interpretado que tiene un lugar especial en los entornos de desarrollo de aplicaciones para Internet por ser código abierto, estar muy bien documentado, ser ligero, multiplataforma y simple de utilizar. Se utiliza entre otras cosas para la programación de páginas Web dinámicas y se destaca por su capacidad de mezclarse con el código HTML.

Entre las características que podemos notar en el lenguaje PHP tenemos que no es lo mismo que un script escrito en otro lenguaje de programación como Perl o C++, pues en vez de escribir un programa con muchos comandos para crear una salida en HTML, escribimos el código HTML con cierto código PHP embebido en el mismo, que producirá cierta salida. El código PHP se incluye entre etiquetas especiales de comienzo y final (<? ... ?>) que nos permitirán entrar y salir del modo PHP.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales. Aunque el desarrollo de PHP está concentrado en la programación de scripts en la parte del servidor, se puede utilizar para muchas otras tareas, en general se puede considerar un lenguaje de propósito general. Su estructura es tipo C, más no requiere de declaración de variables, para el lenguaje todas son cadenas, en caso de ser números las cadenas, éstos se pueden operar aritmética y lógicamente de manera normal. Tiene manejo de archivos locales, se pueden declarar funciones, es posible utilizar el paradigma de programación orientada a objetos y crear clases.

PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies, entre otras acciones. Existen tres campos en los que scripts escritos en PHP son usados:

- **Scripts en la parte del servidor.** Este es el campo más tradicional y el principal campo de trabajo. Se necesitan tres elementos para que esto funcione. El analizador PHP (CGI ó módulo), un servidor Web y un Navegador. Se necesita correr el servidor Web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectando con el servidor Web.

- **Scripts en línea de comandos.** Se pueden crear scripts en PHP y correrlos sin necesitar de un servidor Web ó Navegador. Solamente se necesita el parser PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en Unix ó Linux) ó el Planificador de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesado de texto.
- **Escribir aplicaciones gráficas clientes.** Si se usan adecuadamente las capacidades de PHP, y se utilizan algunas características avanzadas en programas clientes, es posible utilizar PHP-GTK (extensión de PHP) para escribir dichos programas. Es también posible escribir aplicaciones independientes de una plataforma.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape y iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

Así que, con PHP se tiene la libertad de escoger el sistema operativo y el servidor más apropiado o bien disponible. También se tiene la posibilidad de usar programación de procedimientos ó programación orientada a objetos. Aunque no todas las características estándares de la programación orientada a objetos están implementadas en la versión actual de PHP, muchas librerías y aplicaciones grandes (incluyendo la librería PEAR) están escritas íntegramente usando programación orientada a objetos.

Con PHP no está uno limitado a resultados en HTML. Entre las habilidades de PHP se incluyen, creación de imágenes, archivos PDF y películas Flash (usando libswf y Ming) en tiempo de ejecución. También se pueden presentar otros resultados, como XHTML y archivos XML. PHP puede generar estos archivos y grabarlos en el sistema de archivos en vez de presentarlos en la pantalla.

3.3.2 Tecnología ASP

ASP es un desarrollo propio de Microsoft, el cual consiste básicamente en un lenguaje de tratamiento de textos (scripts) basado en Basic y que se denomina VBScripts (Visual Basic Script), se utiliza casi exclusivamente en los servidores web de Microsoft (Internet Information Server y Personal Web Server), los scripts de ASP se ejecutan por lo tanto en el servidor y puede utilizarse conjuntamente

con HTML y Javascript para realizar tareas interactivas en tiempo real con el cliente, con ASP se pueden realizar fácilmente páginas de consulta de bases de datos, funciones sencillas como obtener la fecha y la hora actual del sistema servidor, cálculos matemáticos simples, es decir, el usuario de Internet puede recibir páginas generadas dinámicamente en el servidor.

Agrupadas en la categoría de lenguajes scripts las páginas ASP contienen además de etiquetas de HTML, fragmentos de código que el servidor interpretará antes de enviarlo al navegador. Cabe señalar que tanto ASP, VBScript y Javascript son lenguajes de programación comunes, su sintaxis es implementada en código ASCII por lo que para poder crear, editar y modificar dicho código solo es necesario utilizar un simple y común editor de textos, el cual puede ser el edit de MS-DOS, el Notepad o el Wordpad de los entornos del sistema operativo Windows o cualquiera de los múltiples editores de texto que existen actualmente.

Microsoft ha tenido la diferencia de crear, dentro de su suite de desarrollo DevStudio, una aplicación específica para administración y creación de proyectos Web denominada Visual InterDev. La facilidad para conectar con una base de datos y extraer datos de la misma dinámicamente visualizándolos en el navegador es la utilidad más practicada en las páginas ASP, puede conectarse a gestores de base de datos SQL, Access, Oracle o cualquier otro motor que disponga de driver ODBC.

Para procesar una página ASP no existe ninguna restricción especial en el lado del cliente, por lo que es indiferente la utilización del navegador Internet Explorer o Netscape Communicator sin embargo, en el lado del servidor, es necesario un servidor Web de Microsoft. Se utiliza el archivo ASP.DLL para interpretar el código, siendo el servidor más extendido Internet Information Server. Los servidores de contenidos ASP posibles para plataformas Microsoft son:

- Internet Information Server 3.0 o superior (para Windows NT)
- Personal Web Server (para Windows 95 y Windows 98)

Las páginas ASP comienzan a ejecutarse cuando un usuario solicita un archivo .asp al servidor Web a través del explorador. El servidor web llama a ASP, que lee el archivo solicitado, ejecuta las secuencias de comandos que encuentre y envía los resultados al explorador del cliente. Puesto que las secuencias de comandos se ejecutan en el servidor, y NO en el cliente, es el servidor el que hace todo el trabajo necesario para generar las páginas que se envían al explorador. Las secuencias de comandos quedan ocultas a los usuarios, estos solo reciben el resultado de la ejecución en formato HTML.

Los archivos .asp son archivos de texto normales, no es necesario ningún editor especial para crearlos, puede usarse cualquier editor que genere código ASCII, si no contiene código ASP se comporta como un archivo .html normal. Como sucede en otros lenguajes de programación, en ASP existe una sentencia de

declaración opcional del lenguaje a utilizar, cabe recordar que para identificar las marcas de código ASP se utilizan las etiquetas <% ... %>.

3.3.3 Tecnología JSP

JSP es una especificación de Sun Microsystems cuyo objetivo final es separar la interfaz (presentación visual) de la implementación (lógica de ejecución). Permite crear y gestionar páginas Web dinámicas, ya que se puede mezclar en una página código HTML para generar la parte estática, con contenido dinámico generado a partir de las marcas especiales <% ... %>. En otras palabras, el contenido dinámico se obtiene debido a la posibilidad de incrustar dentro de la página código Java de diferentes maneras.

La tecnología JSP separa la interfase de usuario de la generación de contenidos, permitiendo a los diseñadores cambiar el diseño de las páginas sin alterar el contenido dinámico. Cuando la interacción con el usuario se complica, surge la necesidad de utilizar los componentes Java específicamente diseñados para resolver la lógica del sistema al mismo tiempo que es configurada fácilmente a través de páginas externas HTML. Este tipo de componentes son los llamados Enterprise JavaBeans y las páginas HTML desde las que dichos objetos son ejecutados se llaman páginas JSP.

Cuando un cliente pide este tipo de página, el servidor ejecuta los comandos Java y JavaBeans y distribuye la página resultante al cliente. JSP es la plataforma Java para la construcción de aplicaciones con contenido dinámico web tal como HTML, DHTML, XHTML y páginas XML. El uso de la tecnología JSP ofrece las siguientes ventajas:

- **Separación entre contenidos dinámico y estático:** El modelo JSP permite capturar la lógica de la aplicación en componentes JavaBeans estándar y reutilizables en los que se puede definir la presentación utilizando etiquetas especiales JSP y pequeñas secciones de código Java conocido como scriptlets.
- **Soporte para programación dinámica (scripting):** Las JSP permiten incluir líneas de programación dinámica ejecutadas en el servidor cuando la página es pedida.
- **Una vez escritas, funcionan en cualquier lugar:** Las JSP son una extensión de los Java servlets, un estándar 100% Java de JavaSoft. Las JSP heredan todas las ventajas de la plataforma Java incluyendo “escríbalo una vez y hágalo funcionar en cualquier lugar” o funcionalidad multi-plataforma. Debido a que JSP es un estándar abierto, se puede implementar aplicaciones JSP en cualquier plataforma que se adhiera a su especificación.

- **Rendimiento avanzado y escalable:** Las aplicaciones JSP disfrutan de la misma escalabilidad y rendimiento que los Java servlets ya que se trata de una extensión de la arquitectura Java servlet.
- **Gran calidad de soporte y documentación:** La tecnología JSP es un estándar Java que permite a los usuarios y desarrolladores ordenar toda la documentación en este estándar.

3.3.4 ColdFusion

ColdFusion introdujo el concepto revolucionario de extender al Lenguaje HTML, de tal manera que dentro de su código se incluyesen instrucciones para el manejo de las BD. Reconoce la complejidad del manejo e interacción de scripts CGI, ofreciendo una potente seguridad, veloz carga de datos, procesamiento rápido de scripts CGI que posibilita el cumplimiento de tareas de entrada o devolución de datos. Utiliza fuentes de datos ODBC de 32-bits, las cuales deberán cumplir con el nivel 1 de los ODBC API y soportar las sentencias SQL.

ColdFusion centra su potencialidad en la confiabilidad y el control del manejo de datos. Hace uso de formas HTML estándar con validación de datos de los campos, para realizar la inserción y actualización de registros dentro de una tabla en una base de datos. Para la entrada de datos se especificará el tipo de dato a introducir en un campo específico. Este tipo de datos puede ser de valor entero, flotante, de fecha o en un rango especial de fechas.

Además, se puede registrar la hora de introducción del valor de un campo, la dirección IP (Protocolo Internet) desde la que se hace una solicitud, el nombre del cliente y el tipo de navegador que éste utiliza para acceder los datos, todo ello sin necesidad de escribir una línea de código. En cuanto al despliegue de datos, ColdFusion solamente recibe la solicitud del cliente y realiza la presentación de los mismos de una manera muy sencilla. ColdFusion posee un control completo del formato de despliegue de datos, permite colocar enlaces entre los mismos datos extraídos de la base, en las páginas HTML que han sido generadas al vuelo.

Gracias a esta flexibilidad, se puede realizar cualquier tipo de selección de despliegue de datos, y el software se acomodará a las especificaciones realizadas. Como puede observarse, ColdFusion realiza todo el procedimiento necesario, desde la recolección de información en un servidor de base de datos SQL, hasta el despliegue de la misma. Así mismo, ColdFusion permite alojar procedimientos y pasar datos a ellos muy fácilmente.

Entre las funciones de ColdFusion están:

- Sirve a cualquier requisición de datos una vez que se cuente con la instalación y configuración de las fuentes de datos ODBC de 32-bits.
- Detecta errores producidos por la mala configuración o por el registro completo de la bitácora del servidor SQL.
- Funciona correctamente en una máquina remota. Se ejecuta sin problemas en el Microsoft Internet Information Server (IIS), aún teniendo gran cantidad de solicitudes. Gracias a ello brinda un correcto funcionamiento tanto en Internet como en Intranets.
- Provee de ayuda para la configuración que permita generar páginas HTML en forma dinámica.
- Crea estructuras condicionales dinámicamente para personalizar la solicitud de datos y el envío de los mismos hacia el cliente. Así mismo, diseña cadenas de datos para crear dinámicamente menús desplegables y para llenar listas de selección y listas de documentos.

Para crear aplicaciones de ColdFusion, se necesita del conocimiento previo de sentencias SQL para la generación de código en la selección de la correcta información alojada en una base de datos. Gracias a las sentencias SQL se tiene un control completo sobre qué, dónde y por qué desplegar los datos dentro de un sitio Web.

Funcionamiento de ColdFusion

Una vez se ha realizado la instalación de este paquete, se pueden realizar requisiciones a través de un URL, las cuales son enviadas al servidor Web, y éste a su vez la hace a la interfaz de ColdFusion, la que se conecta a una fuente de datos ODBC, a la cual solicita los datos que requiere extraer de la base de datos. Como puede verse, ColdFusion utiliza fuentes de datos ODBC, de las que incluye una versión dentro del software de instalación, para poder manipular la información dentro de las bases de datos. Una vez se ha obtenido la información que se ha solicitado, la interfaz envía los datos hacia el servidor Web y éste al navegador, en donde los mismos son desplegados gráficamente. En la figura 14 se muestra el proceso que sigue ColdFusion al momento de recibir y responder a una requisición.

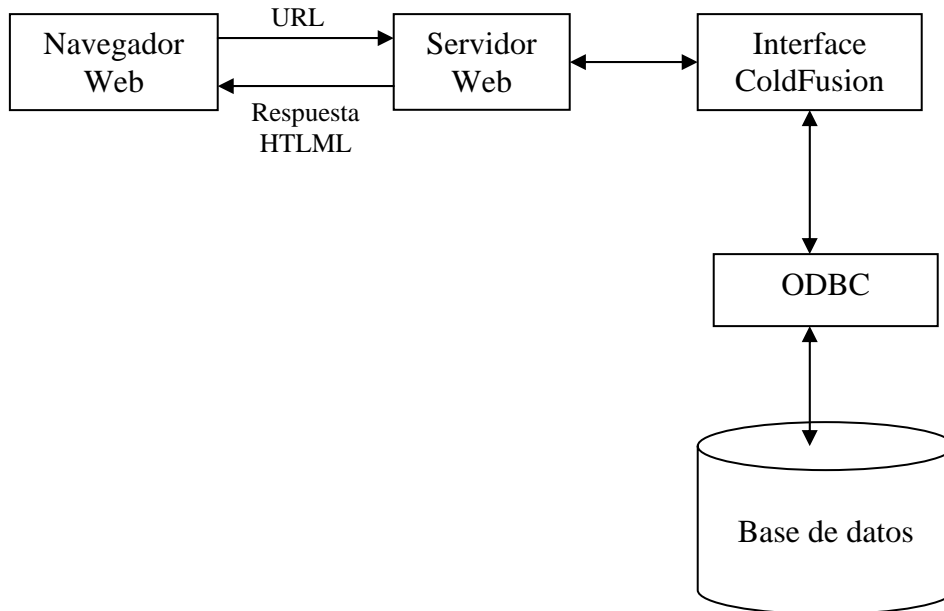


Figura 4. Funcionamiento de ColdFusion

3.3.5 Elección de la Herramienta

JSP y ASP sirven para hacer, más o menos, el mismo tipo de aplicaciones Web. Sin embargo, JSP sigue la filosofía de la arquitectura Java de "escribe una vez ejecuta donde quieras" mientras que la implantación de ASP está limitada para arquitecturas basadas en tecnología Microsoft. JSP se puede ejecutar en los sistemas operativos y servidores Web más populares, como por ejemplo Apache, Netscape o Microsoft IIS. Mientras que ASP sólo tiene soporte nativo para los servidores IIS y Personal Web Server, que son los dos servidores web para sistemas Microsoft, el primero con tecnología NT y el segundo para sistemas Windows 98 y similares.

Por su parte, PHP Es software libre, lo que implica menores costos y servidores más baratos que otras alternativas, a la vez que el tiempo entre el hallazgo de una petición y su resolución es más corto. Además, el volumen de código PHP libre es mucho mayor que en otras tecnologías, siendo superado por Perl, que es más antiguo. Esto permite construir sitios realmente interesantes con sólo instalar scripts libres como PHP Nuke. Es muy rápido, su integración con la base de datos MySQL también es veloz, le permite constituirse como una de las alternativas más atractivas para sitios de tamaño medio-bajo. Su librería estándar es realmente amplia, lo que permite reducir los llamados 'costos ocultos', uno de los principales defectos de ASP. PHP es relativamente multiplataforma. Funciona

en toda máquina que sea capaz de compilar su código, entre ellas diversos sistemas operativos para PC y diversos Unix. El código escrito en PHP en cualquier plataforma funciona exactamente igual en cualquier otra. El acceso a las bases de datos de PHP es muy heterogéneo, pues dispone de un juego de funciones distinto por cada gestor. PHP es suficientemente versátil y potente como para hacer tanto aplicaciones grandes que necesiten acceder a recursos a bajo nivel del sistema como pequeños scripts que envíen por correo electrónico un formulario rellenado por el usuario. Existen menos especialistas en PHP que en ASP en nuestro país.

A diferencia de JSP o CGI basados en C, PHP no requiere un conocimiento exhaustivo del lenguaje de programación. A diferencia de Perl, PHP tiene una sintaxis muy fácil de comprender y a diferencia de ASP, no requiere conocer más de un lenguaje de programación o de la instalación de módulos externos o comerciales para realizar tareas más complicadas no previstas en el lenguaje más usado (Visual Basic Script).

ColdFusion es un producto comercial y su sintaxis limitada impone restricciones haciendo compleja la tarea de agregar funcionalidad a un sitio. Su confiabilidad se ha estropeado con varios problemas de seguridad, que pueden ser mejorados levemente con el uso de una plataforma Unix, sin embargo los costos de licencia son muy altos llegando incluso a superar el costo del hardware. ColdFusion abstrae todo el acceso a base de datos configurando las conexiones fácilmente, llamando simplemente la conexión por nombre sin el conocimiento de con que base de datos se esta conectando. Esta abstracción hace que el acceso a una base de datos específica sea imposible. Aunado a lo anterior, ColdFusion necesita tomar grandes cantidades de tiempo, energía y memoria del CPU.

3.3.6 Requerimientos de seguridad informática que debe de cubrir el sistema

Los requerimientos de seguridad son vitales para nuestro sistema, y consideramos lo siguiente:

1. Solamente se podrá acceder al sistema con un nombre de usuario y una contraseña.
2. La contraseña deberá manejarse de manera cifrada.
3. Cuando se de de alta a un usuario, el sistema deberá de asegurarse de que las contraseñas asignadas sean contraseñas “seguras”.
4. Por tratarse de un sistema de administración de bases de datos el servidor Web trabajará en modo SSL.
5. Para evitar que el usuario vea los errores o las alarmas que pudiera tener nuestro sistema, y de que por ende se muestren datos que algún intruso pudiera utilizar para comprometer el sistema se configurará PHP y Apache, de modo que no se muestren los errores ni las alarmas en las páginas.



4 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se describen las etapas del proceso de planeación y desarrollo de un sistema, mostrando la estructura y componentes de nuestro proyecto.

Diseño e implementación del sistema

4.1 Antecedentes

Ingeniería de Software

Ingeniería de software es la rama de la ingeniería que crea y mantiene las aplicaciones de software usando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos.

La ingeniería de software, como las disciplinas tradicionales de ingeniería, tiene que ver con el costo y la confiabilidad. Algunas aplicaciones de software contienen millones de líneas de código que se espera que se desempeñen bien en condiciones siempre cambiantes.

En un nivel más técnico, la Ingeniería de Software comienza con una serie de tareas que hacen modelos y que resultan en una especificación completa de requisitos y una representación comprensiva de diseño del software que será construido. Se han desarrollado muchos métodos para hacer modelos de sistemas de información. Sin embargo, se prevé que los métodos orientados a objetos podrían llegar a ser el estándar.

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, y por otro lado una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software.

Sin embargo, la forma de agrupar las actividades, los objetivos de cada fase, los tipos de productos intermedios que se generan, etc. pueden ser muy diferentes dependiendo del tipo de producto o proceso a generar y de las tecnologías empleadas.

La complejidad de las relaciones entre las distintas actividades crece exponencialmente con el tamaño, con lo que rápidamente se haría inabordable si no fuera por la vieja táctica de “divide y vencerás”. De esta forma la división de los proyectos en fases sucesivas es un primer paso para la reducción de su complejidad, tratándose de escoger las partes de manera que sus relaciones entre sí sean lo más simples posibles.

La definición de un ciclo de vida facilita el control sobre los tiempos en que es necesario aplicar recursos de todo tipo (personal, equipos, suministros, etc.) al proyecto. Si el proyecto incluye subcontratación de partes a otras organizaciones, el control del trabajo subcontratado se facilita en la medida en que esas partes encajen bien en la estructura de las fases. El control de calidad también se ve facilitado si la separación entre fases se hace corresponder con puntos en los que

ésta deba verificarse (mediante comprobaciones sobre los productos parciales obtenidos).

De la misma forma, la práctica acumulada en el diseño de modelos de ciclo de vida para situaciones muy diversas permite que nos beneficiemos de la experiencia adquirida utilizando el enfoque que mejor se adapte a nuestros requerimientos.

ELEMENTOS DEL CICLO DE VIDA

Un ciclo de vida para un proyecto se compone de **fases sucesivas** compuestas por tareas planificables. Según el modelo de ciclo de vida, la sucesión de fases puede ampliarse con **bucles de realimentación**, de manera que lo que conceptualmente se considera una misma fase se pueda ejecutar más de una vez a lo largo de un proyecto, recibiendo en cada pasada de ejecución aportaciones de los resultados intermedios que se van produciendo realimentación (figura 15).

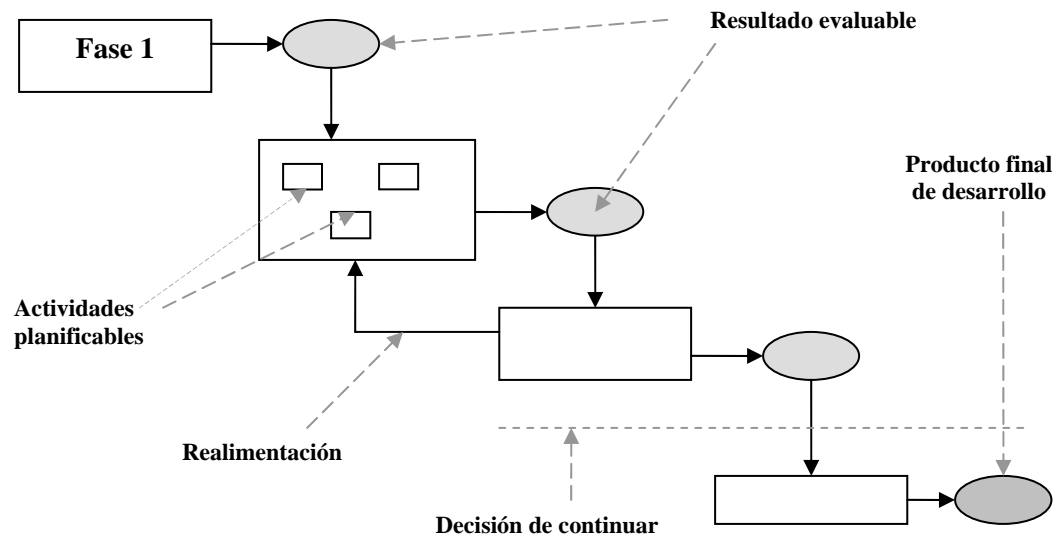


Figura 1. Sucesión de fases

Para un adecuado control de la progresión de las fases de un proyecto se hace necesario especificar con suficiente precisión los resultados evaluables, es decir, productos intermedios que deben resultar de las tareas incluidas en cada fase. Normalmente estos productos marcan los hitos entre fases.

A continuación presentamos los distintos elementos que integran un ciclo de vida:

- Fases. Una fase es un conjunto de actividades relacionadas con un objetivo en el desarrollo del proyecto. Se construye agrupando tareas (actividades elementales) que pueden compartir un tramo determinado del tiempo de vida de un proyecto. La agrupación temporal de tareas impone requisitos

temporales correspondientes a la asignación de recursos (humanos, financieros o materiales).

Cuanto más grande y complejo sea un proyecto, mayor detalle se necesitará en la definición de las fases para que el contenido de cada una siga siendo manejable. De esta forma, cada fase de un proyecto puede considerarse un "micro-proyecto" en sí mismo, compuesto por un conjunto de micro-fases.

Cada fase viene definida por un conjunto de elementos observables externamente, como son las actividades con las que se relaciona, los datos de entrada (resultados de la fase anterior, documentos o productos requeridos para la fase, experiencias de proyectos anteriores), los datos de salida (resultados a utilizar por la fase posterior, experiencia acumulada, pruebas o resultados efectuados) y la estructura interna de la fase (figura 16).

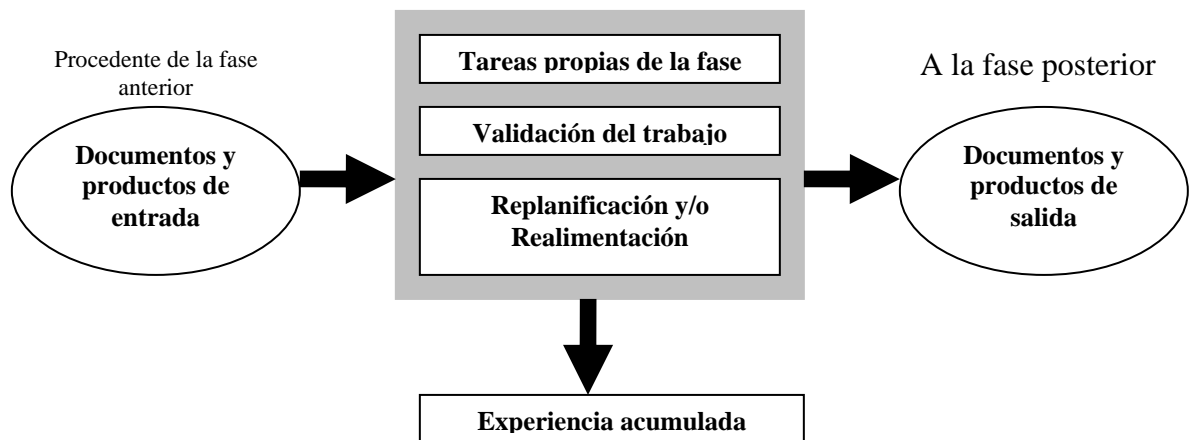


Figura 2. Esquema general de operación de una fase

- Entregables ("deliverables"). Son los productos intermedios que generan las fases. Pueden ser materiales (componentes, equipos) o inmateriales (documentos, software). Los entregables permiten evaluar la marcha del proyecto mediante comprobaciones de su adecuación o no a los requisitos funcionales y de condiciones de realización previamente establecidos. Cada una de estas evaluaciones puede servir, además, para la toma de decisiones a lo largo del desarrollo del proyecto.

TIPOS DE MODELO DE CICLO DE VIDA

Las principales diferencias entre distintos modelos de ciclo de vida son:

- El **alcance** del ciclo dependiendo de hasta dónde llegue el proyecto correspondiente. Un proyecto puede comprender un simple estudio de viabilidad del desarrollo de un producto, o su desarrollo completo o,

llevando la cosa al extremo, toda la historia del producto con su desarrollo, fabricación, y modificaciones posteriores hasta su retirada del mercado.

- Las **características** (contenidos) de las fases en que dividen el ciclo. Esto puede depender del propio tema al que se refiere el proyecto (no son lo mismo las tareas que deben realizarse para proyectar un avión que un puente), o de la organización (interés de reflejar en la división en fases aspectos de la división interna o externa del trabajo).
- La **estructura** de la sucesión de las fases que puede ser lineal, con prototipado, o en espiral. Veámoslo con más detalle:

Ciclo de vida lineal

Es el más utilizado, siempre que es posible, precisamente por ser el más sencillo. Consiste en descomponer la actividad global del proyecto en fases que se suceden de manera lineal, es decir, cada una **se realiza una sola vez**, cada una se realiza **tras la anterior y antes que la siguiente**. Con un ciclo lineal es fácil dividir las tareas entre equipos sucesivos, y prever los tiempos (sumando los de cada fase).

Requiere que la actividad del proyecto pueda descomponerse de manera que una fase no necesite resultados de las siguientes (realimentación), aunque pueden admitirse ciertos supuestos de realimentación correctiva. Desde el punto de vista de la gestión (para decisiones de planificación), requiere también que se sepa bien de antemano lo que va a ocurrir en cada fase antes de empezarla (figura 17).

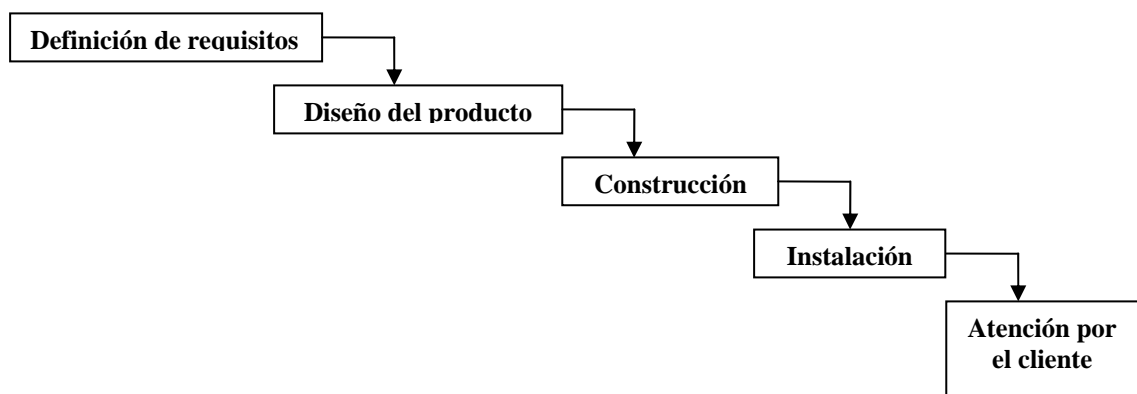


Figura 3. Ejemplo de ciclo lineal para un proyecto de construcción

Ciclo de vida con prototipado

A menudo ocurre en desarrollos de productos con innovaciones importantes, o cuando se prevé la utilización de tecnologías nuevas o poco probadas, que las incertidumbres sobre los resultados realmente alcanzables, o las ignorancias sobre el comportamiento de las tecnologías, impiden iniciar un proyecto lineal con especificaciones cerradas.

Si no se conoce exactamente cómo desarrollar un determinado producto o cuáles son las especificaciones de forma precisa, suele recurrirse a definir especificaciones iniciales para hacer un **prototipo**, o sea, un producto parcial (no hace falta que contenga funciones que se consideren triviales o suficientemente probadas) y provisional (no se va a fabricar realmente para clientes, por lo que tiene menos restricciones de coste y/o prestaciones). Este tipo de procedimiento es muy utilizado en desarrollo avanzado.

Ciclo de vida en espiral

El ciclo de vida en espiral puede considerarse como una generalización del anterior para los casos en que **no basta con una sola evaluación de un prototipo** para asegurar la desaparición de incertidumbres y/o ignorancias. El propio producto a lo largo de su desarrollo puede así considerarse como una sucesión de prototipos que progresan hasta llegar a alcanzar el estado deseado. En cada ciclo (espirales) las especificaciones del producto se van resolviendo paulatinamente.

A menudo la **fuerza de incertidumbres es el propio cliente**, que aunque sepa en términos generales lo que quiere, no es capaz de definirlo en todos sus aspectos sin ver como unos influyen en otros. En estos casos la evaluación de los resultados por el cliente no puede esperar a la entrega final y puede ser necesaria repetidas veces.

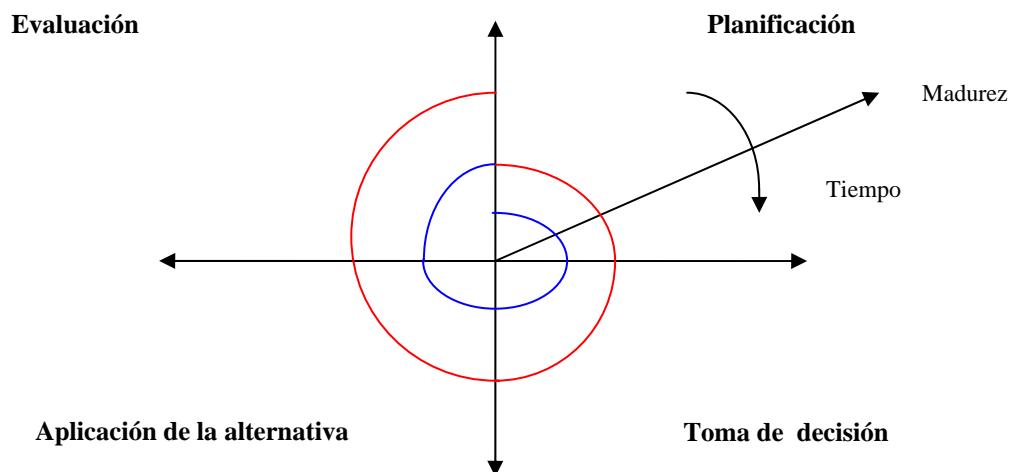


Figura 4. Ciclo de vida en espiral

El esquema del ciclo de vida para estos casos puede representarse por un bucle en espiral (figura 18), donde los cuadrantes son, habitualmente, fases de **especificación, diseño, realización y evaluación** (o conceptos y términos análogos).

En cada vuelta el producto gana en “madurez” (aproximación al final deseado) hasta que en una vuelta la evaluación lo apruebe y el bucle pueda abandonarse.

OBJETIVOS DE CADA FASE

Dentro de cada fase general de un modelo de ciclo de vida, se pueden establecer una serie de objetivos y tareas que lo caracterizan.

Fase de definición (¿qué hacer?)

- Estudio de **viabilidad**.
- **Conocer los requisitos** que debe satisfacer el sistema (funciones y limitaciones de contexto).
- Asegurar que los **requisitos son alcanzables**.
- Formalizar el **acuerdo** con los usuarios.

Fase de diseño (¿cómo hacerlo? Soluciones en coste, tiempo y calidad)

- Identificar **soluciones tecnológicas** para cada una de las funciones del sistema.
- Asignar **recursos** materiales para cada una de las funciones.
- Proponer (identificar y seleccionar) **subcontratas**.
- Establecer métodos de **validación** del diseño.

Fase de construcción

- Generar el producto o servicio pretendido con el proyecto.
- Integrar los elementos subcontratados o adquiridos externamente.

Validar que el producto obtenido satisface los requisitos de diseño previamente definidos y realizar, si es necesario, los ajustes necesarios en dicho diseño para corregir posibles lagunas, errores o inconsistencias.

Fase de mantenimiento y operación

- **Operación:** asegurar que el uso del proyecto es el pretendido.
- **Mantenimiento** (nos referimos a un mantenimiento no habitual, es decir, aquel que no se limita a reparar averías o desgastes habituales -este es el caso del mantenimiento en productos software, ya que en un programa no cabe hablar de averías o de desgaste).

4.2 Diseño e implementación del sistema

Para el diseño de nuestro sistema nos basamos en la ingeniería de software y en el ciclo de vida lineal porque es el ciclo que mejor tiene definidos los pasos a seguir para la aprobación de un producto (figura 19):

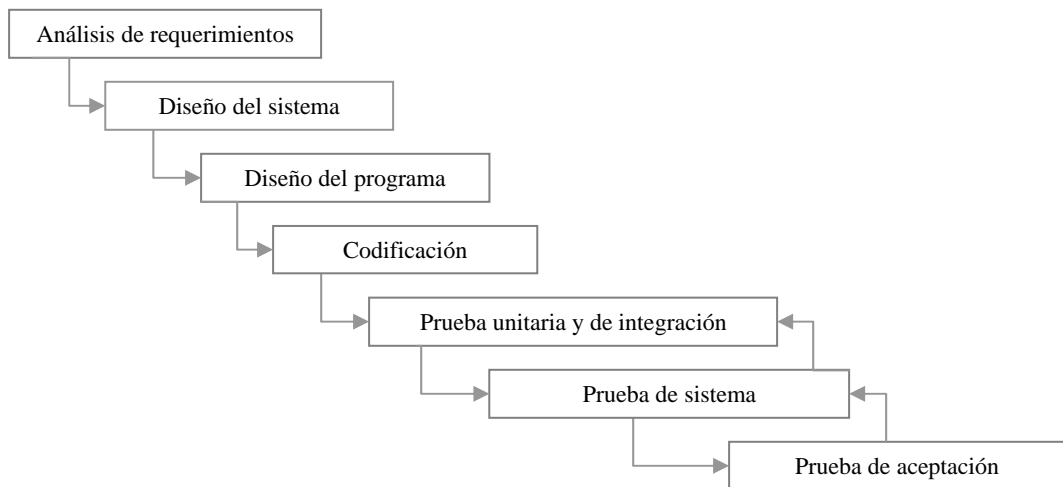


Figura 5. Ciclo de vida para el modulo de administración de bases de datos

En este ciclo por lo regular sigue operación y mantenimiento, pero en este caso nosotros no lo consideramos porque no sabemos si seguiremos trabajando el sistema. En realidad en cada etapa regresamos a otras fases hasta que obtuvimos el sistema deseado, de acuerdo a los requerimientos deseados del sistema. En las sucesivas páginas se describirá lo planeado y concluido en cada parte de nuestro sistema.

- **Fase de análisis de requerimientos.**

En esta fase analizamos como solución al problema de administración de bases de datos un módulo dinámico de administración de bases de datos con software libre y los requerimientos definidos son los siguientes:

- 1.- Crear bases de datos
- 2.- Controlar los accesos a las bases de datos
- 3.- Monitorear el tamaño, y el comportamiento de las base de datos
- 4.- Acceso desde cualquier parte para la administración de las bases de datos.

- **Fase de diseño del sistema**

Lo primero por definir fue el sistema general (Figura 19), ya que era necesario considerarlo común para cualquier red de administración de bases de datos.

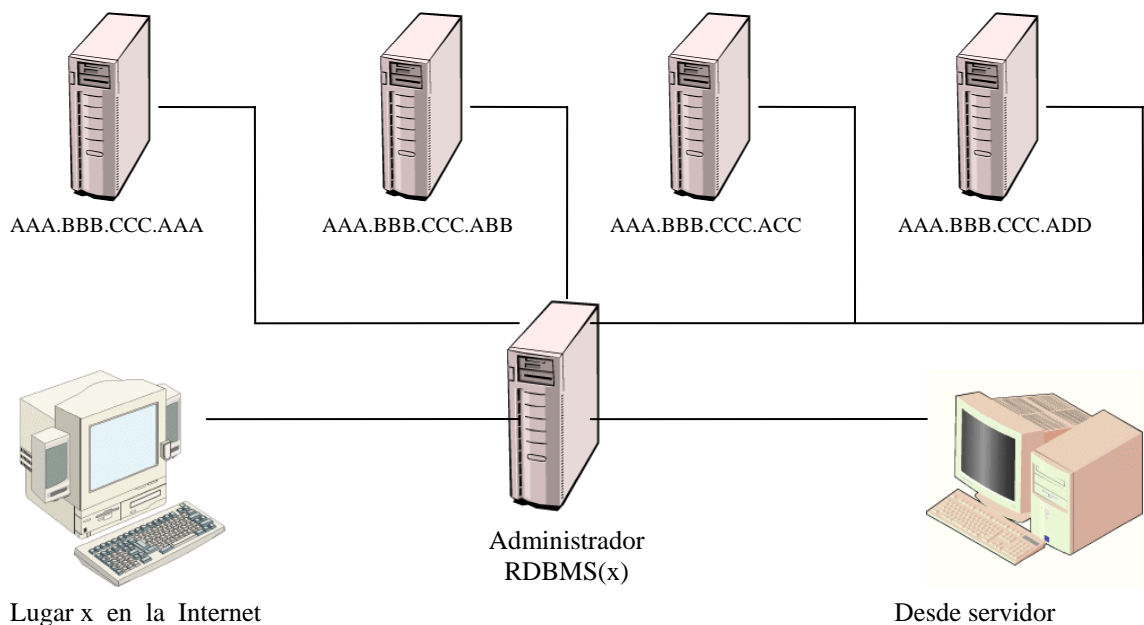


Figura 6. Red general de desarrollo del proyecto

Como podemos ver (Figura 20) se puede administrar el sistema desde cualquier punto de la Internet o desde el servidor para entrar a cualquier otro servidor y acceder a los RDBMS para trabajar con las bases de datos.

Lo primero que definimos fue el como se resolvería el problema de administrar diversos RDBMS y como debería estar enfocado el sistema para ser utilizado por administradores de bases de datos, por lo que se determino que el sistema quedaría formado por las siguientes partes:

- 1.- Acceso al sistema. En esta parte el usuario introducirá sus datos para poder acceder al sistema y poder interactuar con este de acuerdo a los permisos que se le otorguen.

- 2.-Administración de bases de datos. En esta parte se administrará la base de datos, y perfil de los usuarios.
- 3.- Administración del sistema. Esta sección se divide en administración de: usuarios, servidores y RDBMS.
- 4.- Monitoreo. Mediante bitácoras se podrá monitorear el tamaño de la base de datos.

Las partes de acceso a el sistema, monitoreo y registro de usuario representan algunos de los puntos más importantes de este sistema.

Para proporcionar la facilidad de permitir el acceso al sistema desde cualquier parte de la Internet se decidió emplear una aplicación Web, por ser mas cómodo y amigable para el usuario.

Para que el sistema Web reconozca los servidores, los RDBMS con los que trabajará y los usuarios que pueden tener acceso al sistema utilizaremos una base de datos que llamaremos Administrator¹.

No fue necesario desarrollar una aplicación específica para el uso del administrador, es decir, el administrador utiliza el sistema para administrar el sistema, pero el personal que labora decide para que utilizara esta aplicación, ya sea para páginas Web o para otro tipo de aplicaciones de base de datos. Es debido a eso que solo se administra bases de datos, se procuro que el sistema fuera lo más práctico posible y poder colaborar un poco con el desarrollo de sistemas.

El diagrama siguiente (Figura 20) nos muestra las partes que dividen nuestro sistema mediante en un diagrama jerárquico:

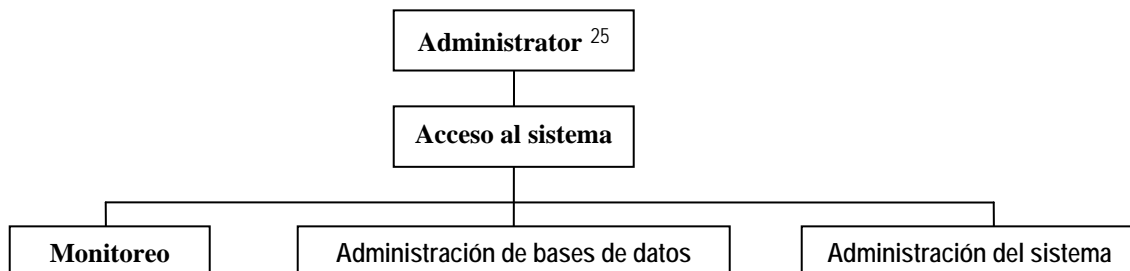


Figura 7. Componentes del sistema

Acceso al sistema

El acceso al sistema (Figura 21) solamente preguntará al usuario su nombre de usuario, contraseña, a que servidor quiere acceder y cual RDBMS usará.

¹ Administrator es el nombre de la base de datos y es el nombre del sistema

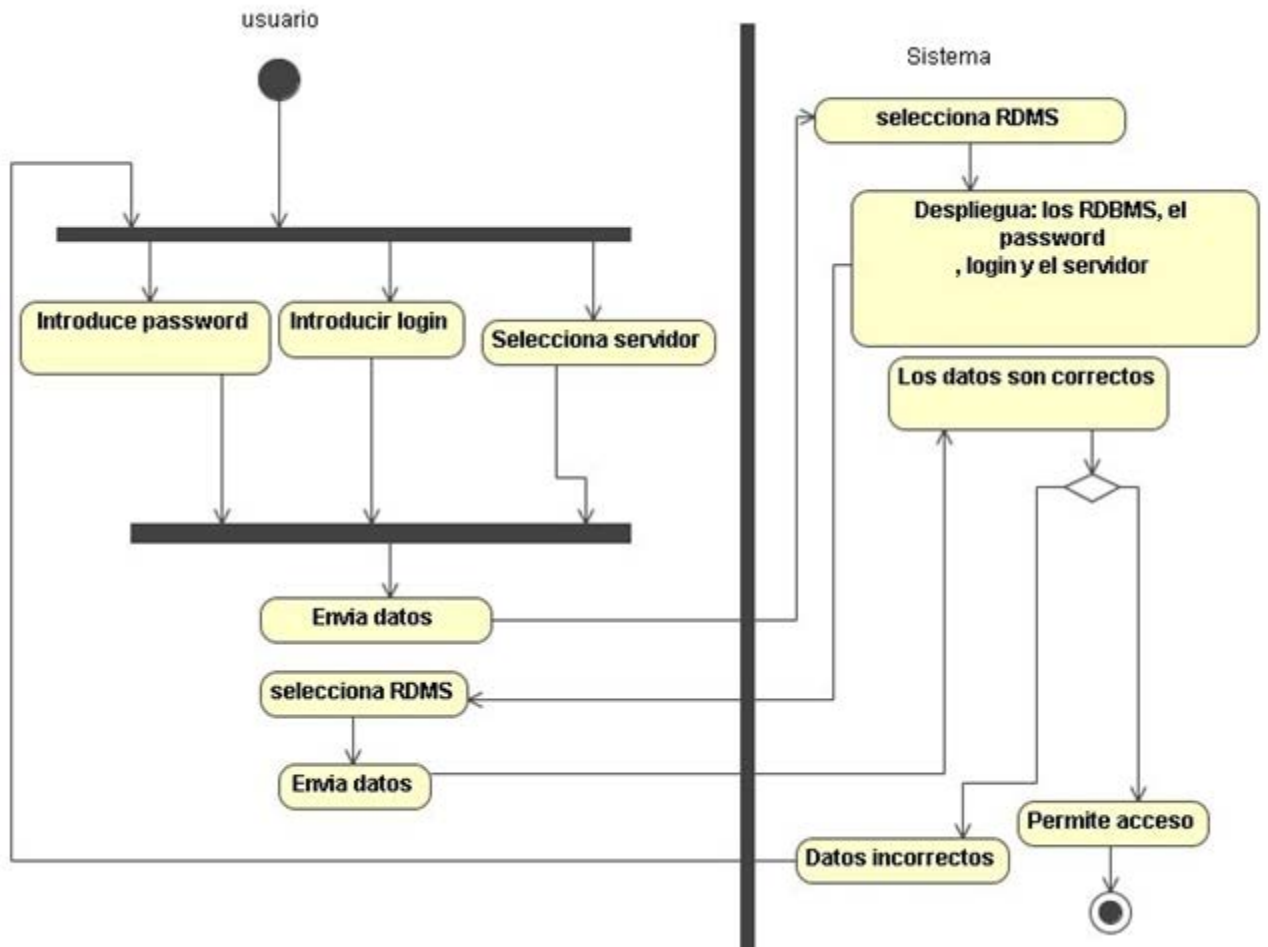


Figura 8. Acceso al sistema

Para interpretar el diagrama anterior hace falta estudiar **UML**²; ya que el diagrama utilizado es un diagrama de actividades, el cual es una forma de representar el comportamiento que se requiere del sistema. En este caso lo que describimos fue el funcionamiento que queremos que tenga el sistema cuando el usuario deba autenticarse al acceder al sistema.

A continuación voy a explicar el diagrama:

1. El usuario introduce el login, contraseña y selecciona el servidor al que desea acceder.
2. El sistema recibe la petición y selecciona los RDBMS existentes, después despliega los RDBMS existentes, la contraseña y el login introducidos por el usuario, y el servidor seleccionado.
3. El usuario selecciona el RDBMS al que desea acceder y envía todos los datos.
4. El sistema verifica si el usuario tiene derecho a acceder al sistema, si tiene permiso permite el acceso, en caso contrario niega el acceso.

² UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

En el siguiente diagrama de actividades (Figura 22) se muestra el procedimiento que realizará Administrator para cada operación que se realizará con las bases de datos, es decir, la manera en que interactúa con los usuarios y otros servidores:

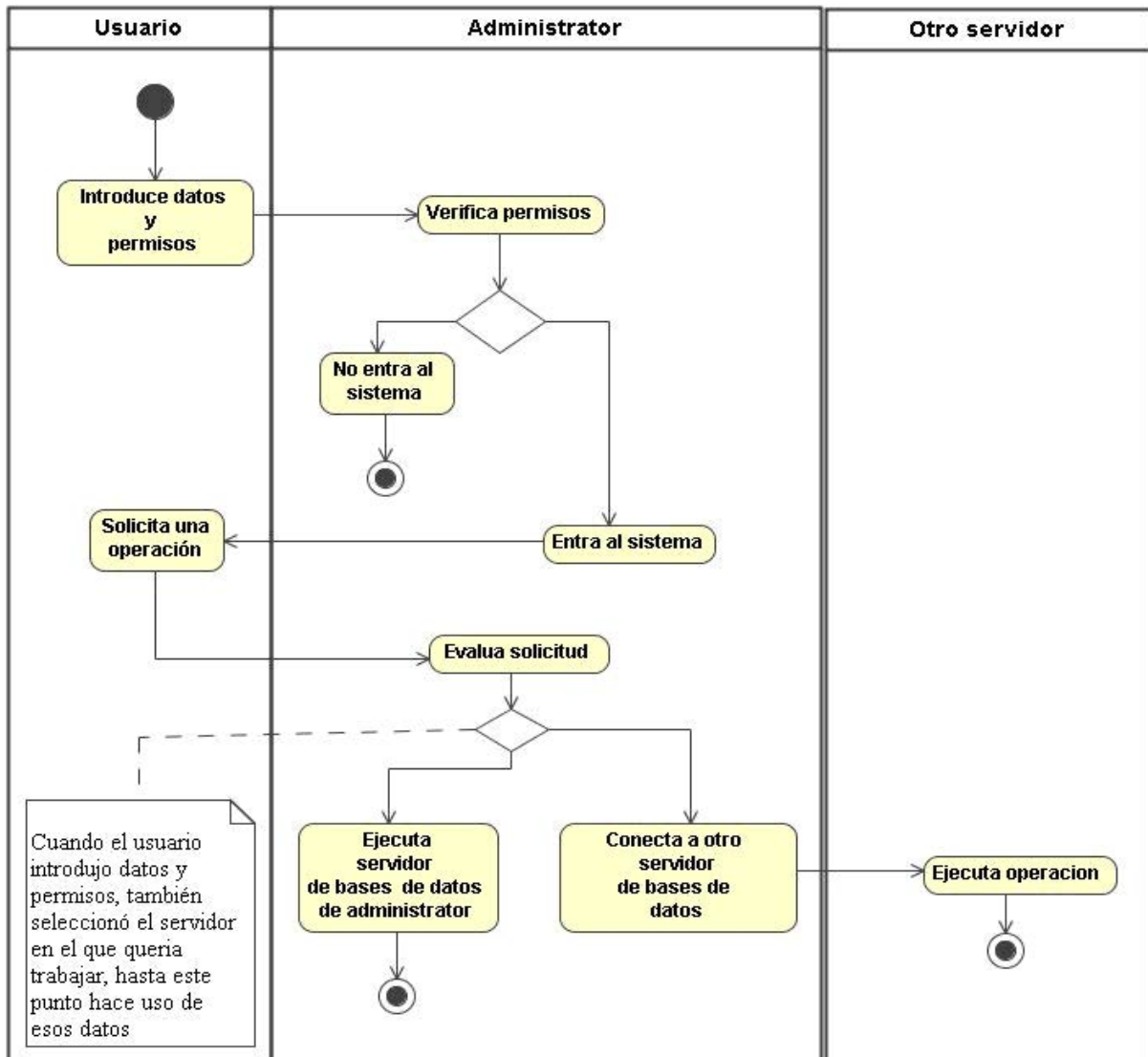


Figura 9. Comportamiento general del sistema

Administración de bases de datos

Esta parte está formada por los siguientes componentes (Figura 24):

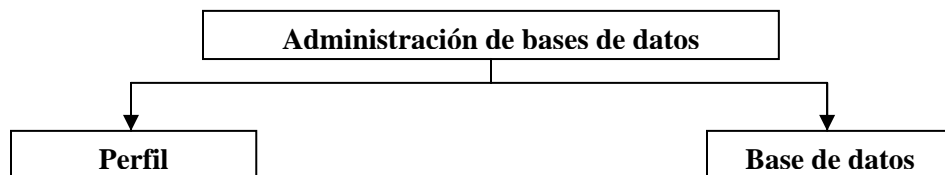


Figura 10. Sección de administración del sistema

En la parte definida como perfil se podrá modificar datos o login y contraseña de los usuarios. En la parte definida como base de datos se podrá crear, borrar, o modificar bases de datos. El funcionamiento deseado para la parte de base de datos se muestra en el diagrama de actividades de la Figura 25.

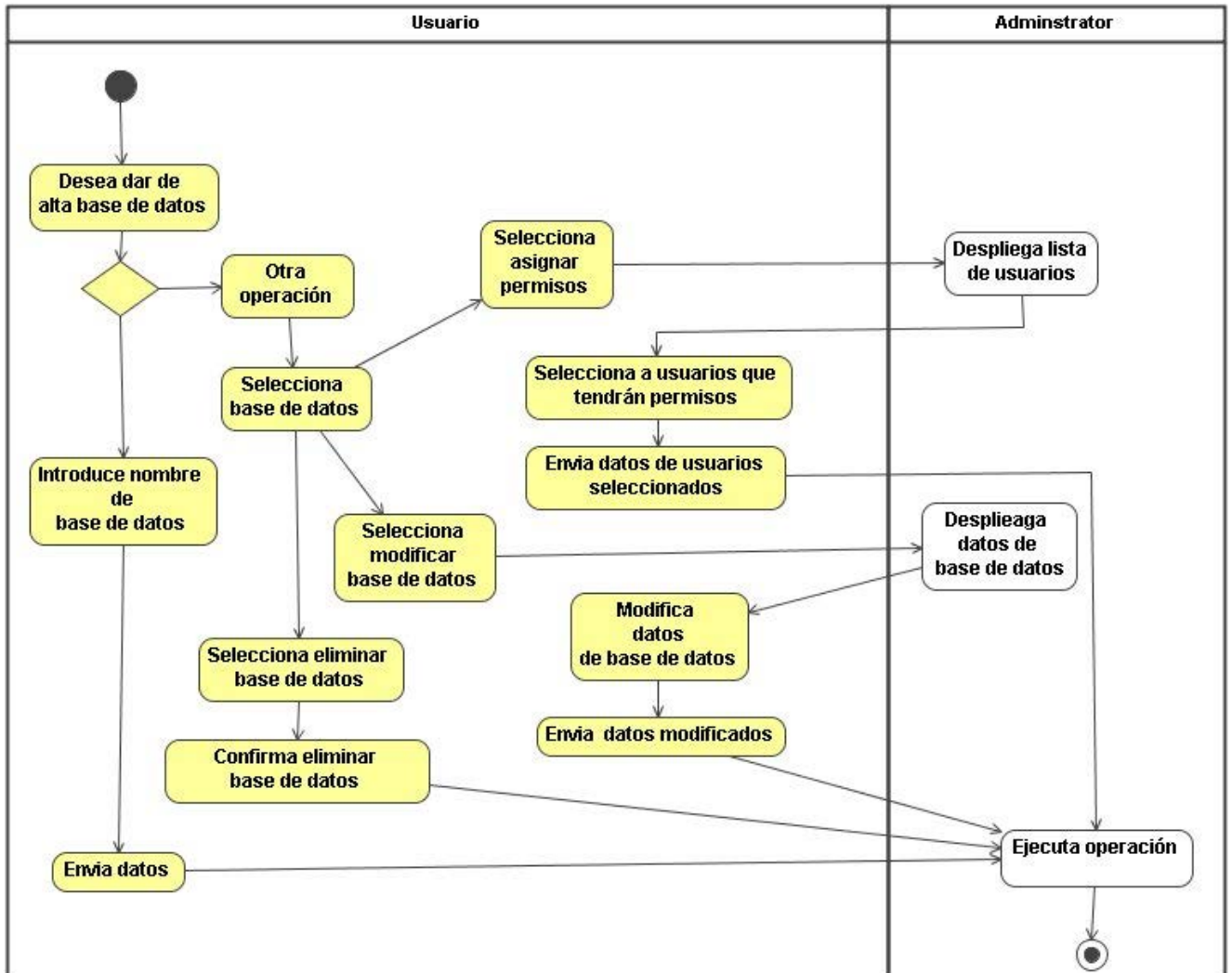


Figura 11. Procedimiento para dar de alta base de datos

Explicación del anterior diagrama (Figura 25):

- 1.- El usuario selecciona la operación que desea realizar, dar de alta, modificar o eliminar una base de datos.
- 2.- Si desea dar de alta la base de datos introduce el nombre de la base de datos y solicita al sistema que de alta la base de datos.

3.- Si desea realizar otra operación, verifica la existencia de la base de datos, selecciona la base de datos.

4.- Selecciona modificar o eliminar base de datos, se despliega la información de la base de datos y el usuario confirma eliminar base de datos o modifica datos de la base de datos.

Monitoreo

La parte de monitoreo del sistema (Figura 26) estará compuesta por tamaño de la bases de datos.

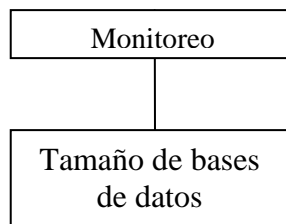


Figura 12. Sección de monitoreo del sistema

Como su nombre lo indica, la sección del tamaño de la base de datos calcula el tamaño de la base de datos, nos despliega el tamaño que ocupa cada tabla de base de datos y nos da el total del tamaño de ésta, monitorear el tamaño de la base de datos nos sirve para que nos podamos darnos cuenta de cuanto espacio ocupan todas y nos da oportunidad de notar si una base de datos ocupa mas espacio que otra con mas prioridad para el usuario.

Administración del sistema

Esta parte está definida por el siguiente diagrama (Figura 27).

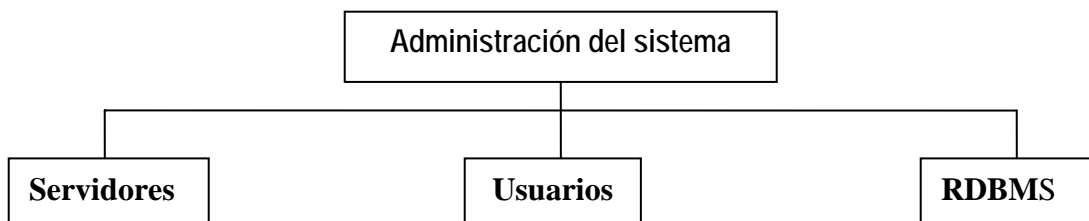


Figura 13. Sección de seguridad del sistema

Esta parte es la mas relevante de todo, debido a que contiene los datos más importantes del sistema, es por ello que solamente los administradores del sistema tienen acceso. Es aquí donde se dan de alta los servidores, los RDBMS y los usuarios que podrán acceder al sistema, mediante un diagrama (Figura 27) se describirá el funcionamiento de la parte de servidores:

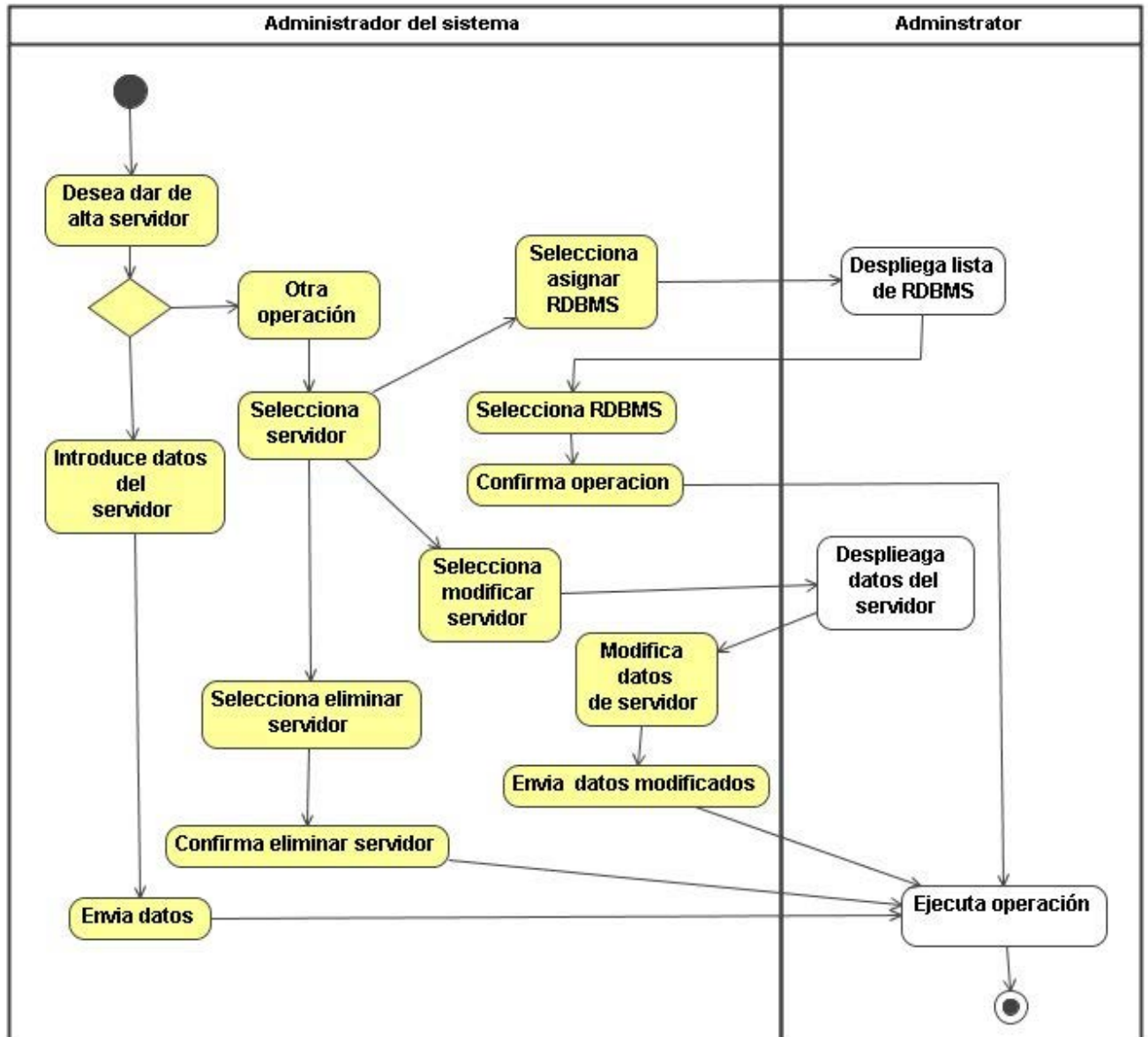


Figura 14. Funcionamiento del módulo de servidores

Explicación del funcionamiento de la sección de servidores:

- 1.- El usuario selecciona la operación que desea realizar, dar de alta, modificar o eliminar servidor.
- 2.- Si desea dar de alta servidor, el usuario introduce IP y nombre del servidor, y envía datos.
- 3.- Si desea realizar otra operación, selecciona el servidor y tiene opción a varias opciones asignar RDBMS, modificar o eliminar servidor.
- 4.- Si selecciona asignar RDBMS, Administrator despliega la lista de los RDBMS, el usuario selecciona los RDBMS y envía los datos.
- 5.- Si selecciona modificar datos de servidor, Administrator despliega los datos del servidor y el usuario modifica los datos.

6.- Si selecciona eliminar servidor, Administrator despliega un mensaje de si desea seguir y el usuario confirma o cancela la operación.

Mediante un diagrama se describirá el funcionamiento de la parte de usuarios (Figura 29):

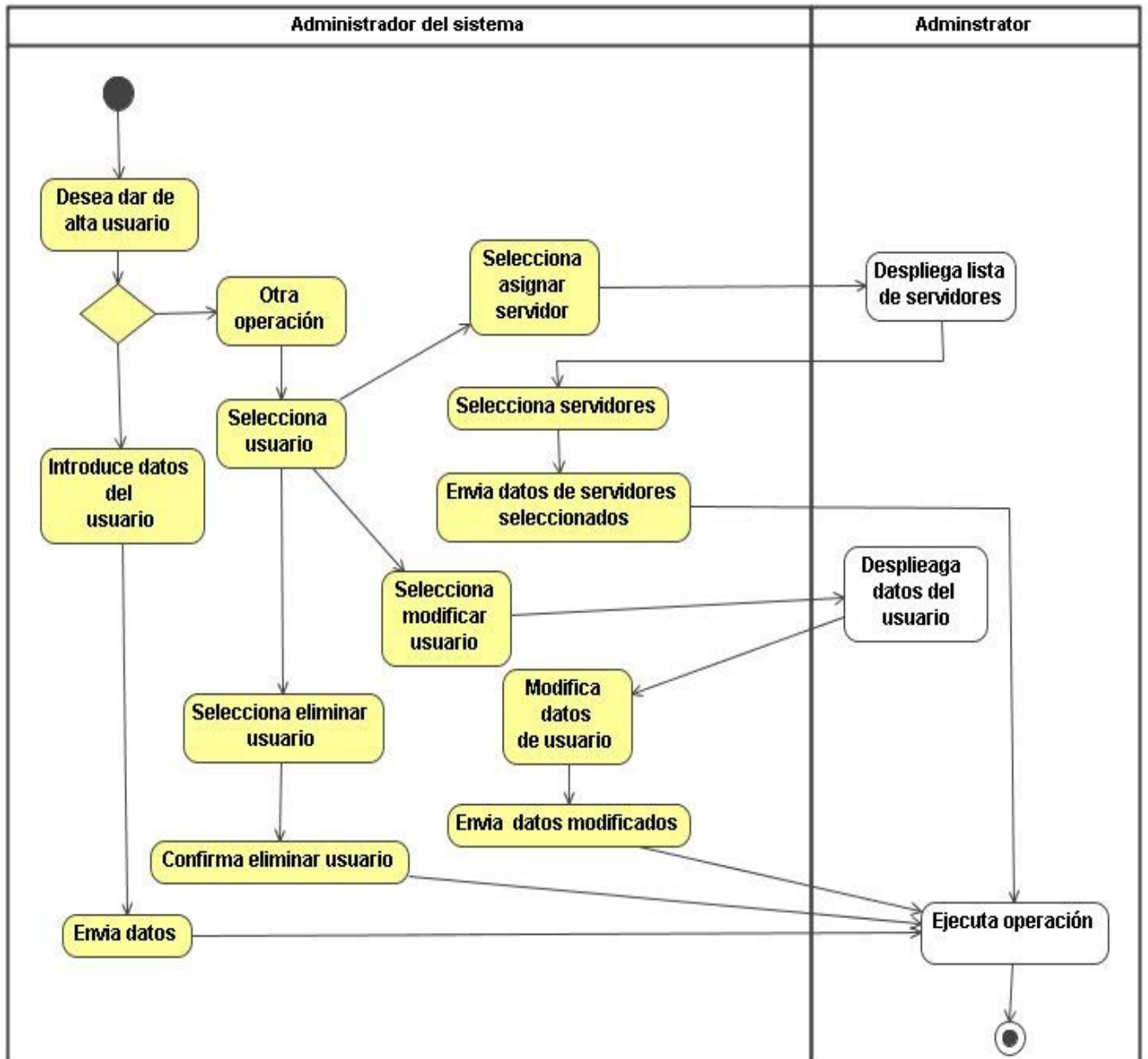


Figura 15. Funcionamiento del módulo de usuario

El diagrama de actividades anterior (Figura 29) nos muestra la forma en que queremos que funcione la parte de usuario, la forma en la que queremos dar de alta a los usuarios, que tendrán acceso al sistema.

El funcionamiento deseado para el RDBMS se muestra en el siguiente diagrama (Figura 30):

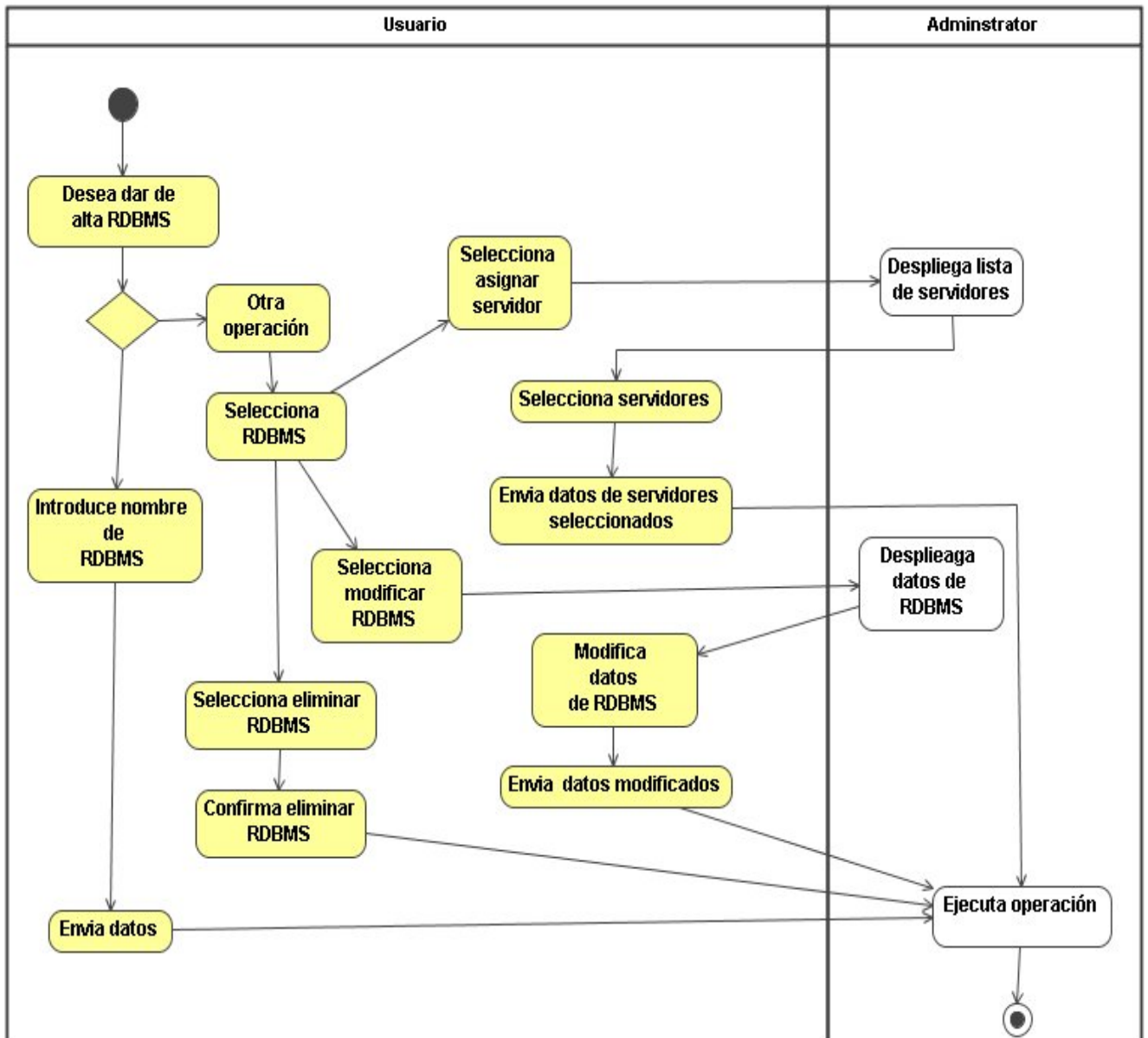


Figura 16. Funcionamiento del modulo de RDBMS

El diagrama anterior (Figura 30) nos muestra la forma en la que funcionará nuestro RDBMS, la forma en que haremos las altas, bajas, modificaciones y

asignación de servidores. En esta sección nosotros le indicamos al sistema que RDBMS hay en los servidores.

A continuación mediante una serie de pasos describiremos el diagrama anterior:

1. El usuario selecciona la operación que desea realizar, dar de alta, modificar, o eliminar RDBMS, o asignar RDBMS a servidores usuario.
2. Si desea dar de alta RDBMS introduce nombre de RDBMS y envía datos
3. Si desea hacer otra operación el usuario selecciona al RDBMS y posteriormente la operación que desea realizar.
4. Si selecciona modificar datos de RDBMS Administrator despliega la información referente a este y el usuario modifica los datos.
5. Si selecciona eliminar RDBMS, Administrator despliega un mensaje de si desea seguir y el usuario confirma o cancela la operación.
6. Si selecciona asignar RDBMS a servidores Administrator despliega la lista de servidores el usuario selecciona los servidores que tienen funcionando ese RDBMS.

Fase de diseño del programa

1.- Diseño gráfico y HTML

En la parte de diseño de las páginas Web del sistema se programo la forma genérica en que aparecerían las páginas Web. En forma general quedo de la siguiente manera (Figura 31):

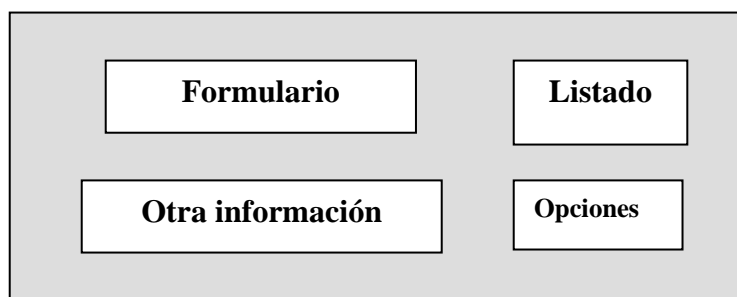


Figura 17. Plantilla de las páginas Web

En la parte de formulario se desplegaran los campos del sistema, en listado se desplegara un listado de los datos que se estén tratando, en otra información se desplegara información importante, y en opciones se podrá solicitar alta, baja, modificación u otras operaciones a los campos del sistema.

La página principal será mediante la que ingrese el usuario a nuestro sistema, y quedara de la siguiente manera (Figura 32):

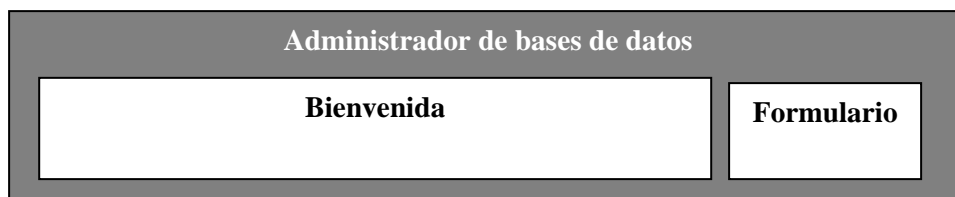


Figura 18. Plantilla de la página principal

En la sección de bienvenida (Figura 31) se desplegará una breve descripción del objetivo de nuestro proyecto y los créditos, y en formulario se desplegarán los campos necesarios para que los usuarios puedan acceder al sistema.

Los menús de administración de bases de datos, monitoreo y administración de sistema se diseñarán de la siguiente forma genérica (Figura 33):

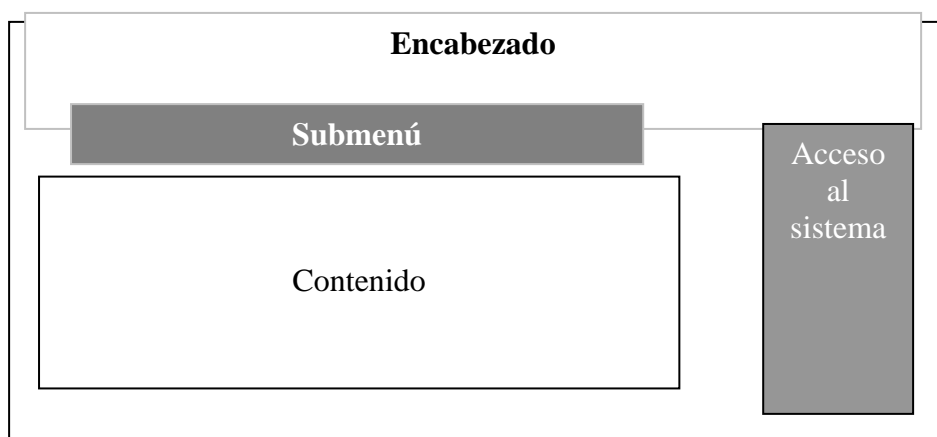


Figura 19. Plantilla para los módulos de administración, monitoreo y seguridad

El encabezado estará formado por las imágenes de nuestro sistema, submenú nos dará las ligas a otras secciones del sistema, acceso al sistema contendrá el formulario que nos dará acceso a otro servidor y por último la sección de contenido nos servirá para poder desplegar las partes referentes a perfil, base de datos, etc. y tendrá una liga para poder ver los registros de esa sección y otra que se llamará trabajar que nos servirá para poder modificar los datos.

2.- Creación de la base de datos.

Después de quedar definidas las páginas HTML, se creó el diagrama entidad relación de la base de datos que manejaremos para este sistema, esta base de datos tiene como fin proporcionarle al sistema los datos necesarios para que pueda reconocer su entorno. Por lo tanto los datos que necesita para poder

trabajar son los datos de los servidores, los RDBMS que se encuentran en estos, y los datos de los usuarios para poder controlar el acceso.

3.- Diseño de los scripts

Como anteriormente vimos nuestro sistema esta dividido en cuatro partes, acceso al sistema, administración de bases de datos, administración del sistema y monitoreo. Antes de mostrar como procedimos con cada subsistema de nuestro sistema es necesario hacer notar a lo que nos enfrentamos en este desarrollo y como fue resuelto.

Debido a que PHP utiliza diferentes funciones para conectarse a todos los RDBMS que soporta y a que todos los RDBMS tienen algunas variantes en su SQL, decidimos resolver el problema con clases, es decir para cada RDBMS soportado en PHP se creó una clase que ejecuta un query, y realiza otros procedimientos. Para lograr manejar esto como regularmente se hace en PHP con aplicaciones para PostgreSQL, MySQL, etc., se creó otra clase llamada general que sirve como punto de conexión con las otras clases, es decir cada clase tiene los mismos métodos, y obviamente la clase general también usa los mismos procedimientos. Es decir se utiliza polimorfismo. A continuación se muestra el diagrama de clases (Figura 34) para conexión a RDBMS, es decir el polimorfismo que utilizamos para este caso.

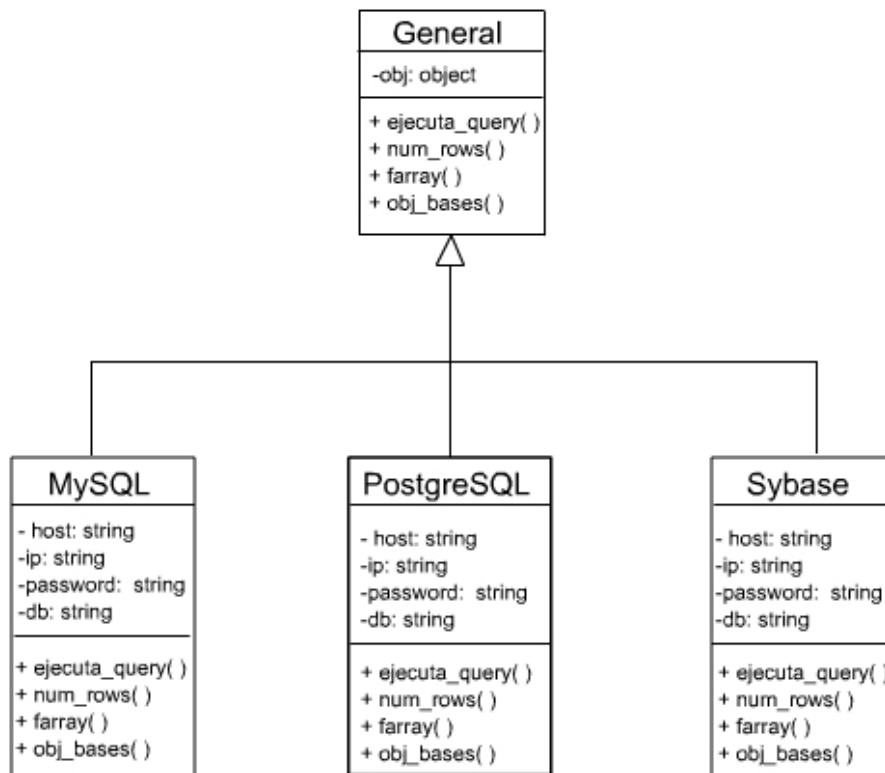


Figura 20. Diagrama de clases

En este caso solo declaramos algunas clases para ejemplificar como fue resuelto el problema.

Como vimos en el diagrama anterior (Figura 33) las clases MySQL, PostgreSQL, y Sybase tienen las mismas variables y funciones, y clase general tiene los mismos procedimientos lo que significa que mediante la clase general nosotros podemos llamar a cualquier procedimiento de cualquier clase, simplificando así mucho nuestro código.

Fase de codificación

En la codificación lo primero que se hizo fue la programación en HTML y posteriormente en PHP. Las secciones quedaron de la siguiente manera:

En la parte principal de la página (Figura 35) se muestra un mensaje de bienvenida con el nombre del sistema y el formulario para poder entrar al sistema, en cuanto se logra el acceso se despliega una página (Figura 36) con las opciones de administración del sistema.

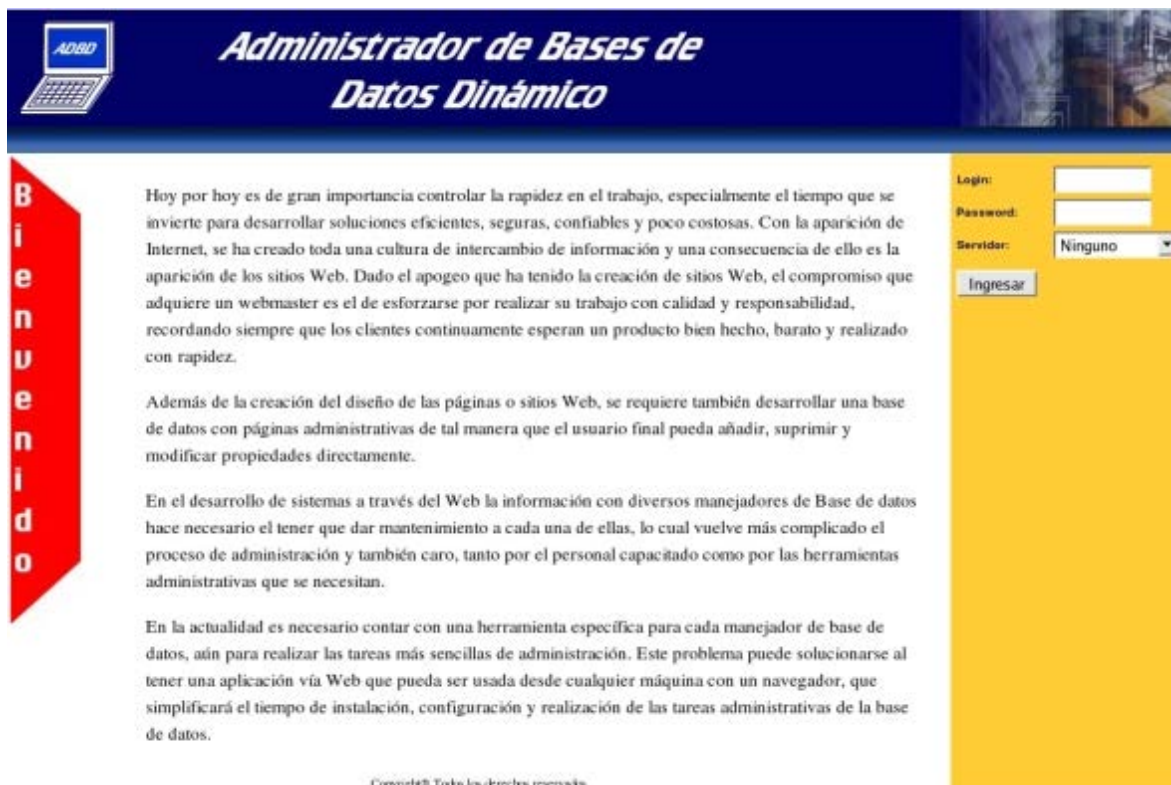


Figura 21. Página principal del sistema

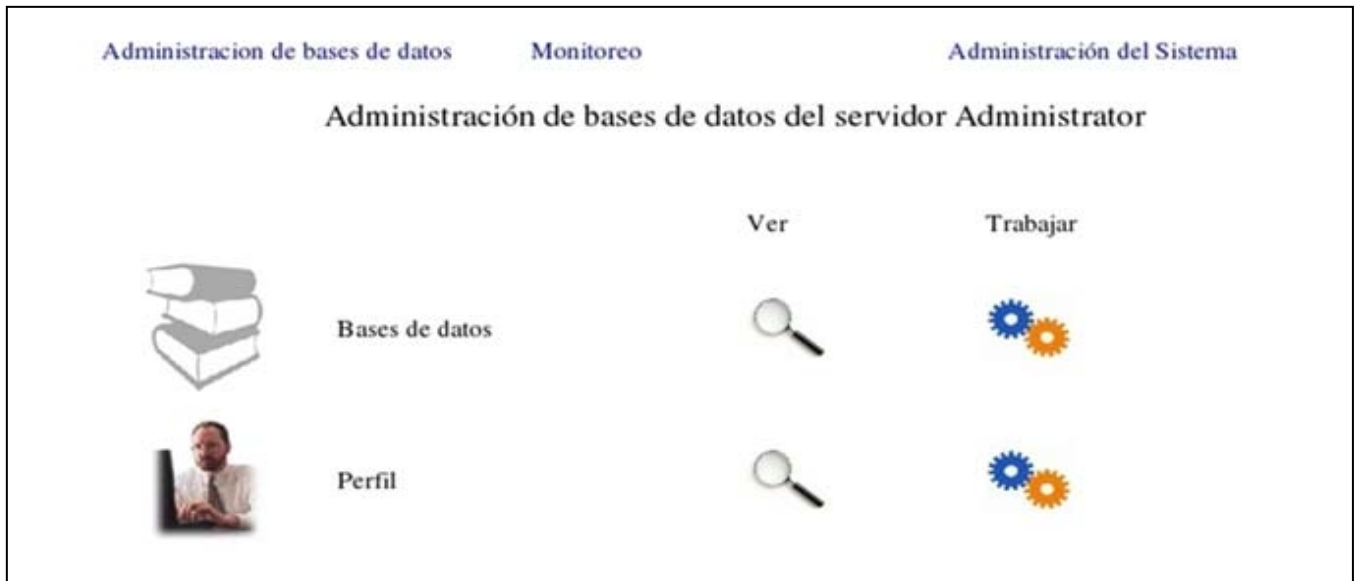


Figura 22. Administración de bases de datos

La sección de administración de RDBMS quedo de la siguiente forma (Figura 37):

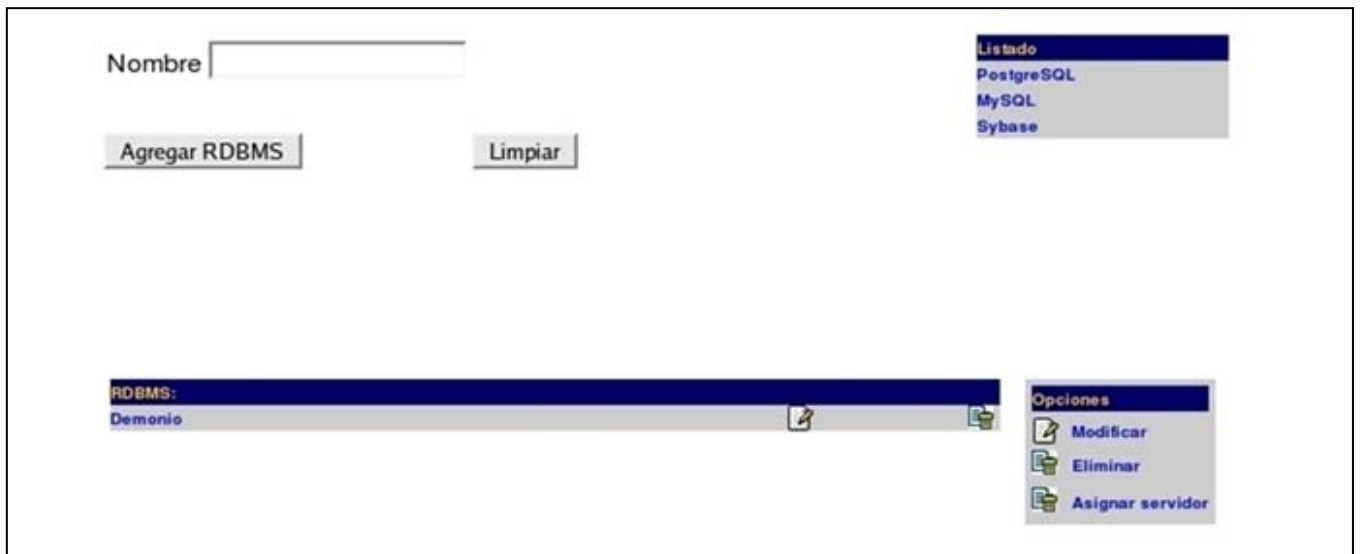


Figura 23. Administrador de RDBMS

Tal como se planeó (Figura 31) en la fase de diseño, las páginas administrativas quedaron divididas en formulario, listado, opciones, y otros datos, para darle al administrador mayor disponibilidad de los datos.

Como podemos notar en la Figura 37 y corroborando lo dicho anteriormente el usuario tiene a la vista todos los RDBMS que maneja el servidor y en la parte

baja izquierda podemos ver un menú que nos muestra la lista de servidores que maneja Administrator, seleccionando cualquiera se modifica la parte de listado y nos muestra los RDBMS de ese servidor, al seleccionar la liga de asignar servidor se abre un menú y es necesario indicarle al sistema que RDBMS están en ese servidor, para que administrador pueda trabajar con ellos. En la parte superior hay un menú para que el usuario pueda acceder a la sección Monitoreo o Administración de bases de datos.

La sección de administración del sistema quedó de la siguiente forma (Figura 38):

Administración de bases de datos Monitoreo Administración del Sistema

Administración del sistema Administrator

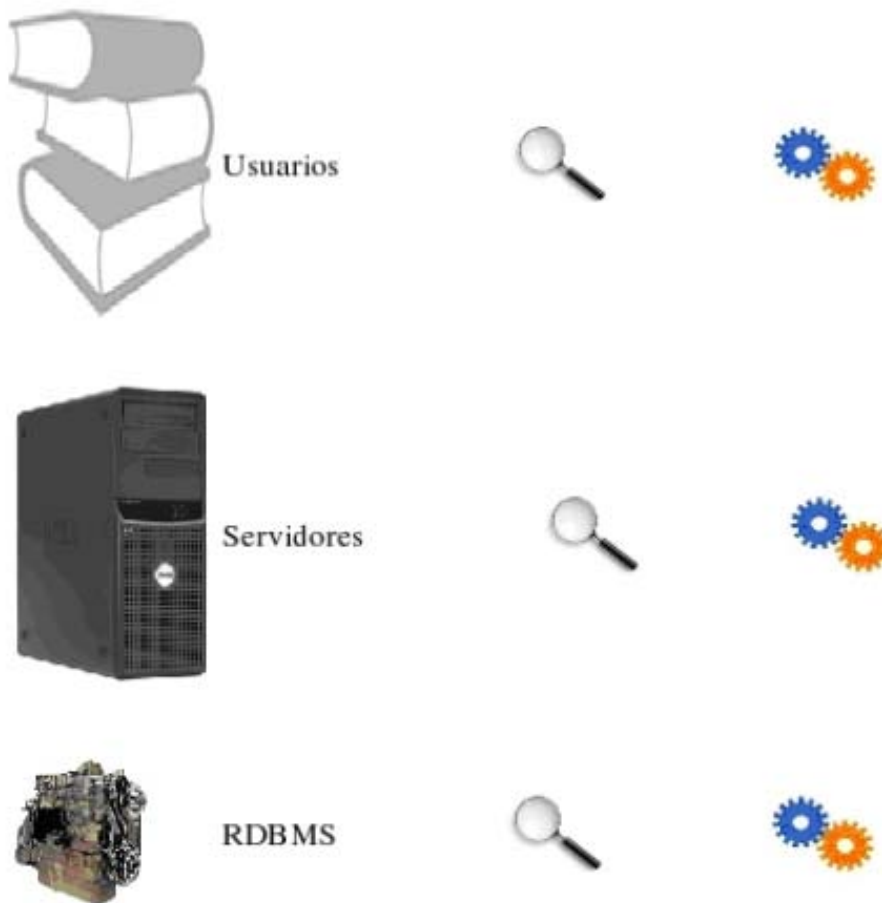


Figura 24. Panel de administración del sistema

Esta sección (Figura 38) es la más importante de todas; porque contiene los datos que pueden comprometer la seguridad de las bases de datos y los servidores.

Programación de los scripts.

Mediante un diagrama de paquetes se describirá como quedo dividido el sistema (Figura 39):

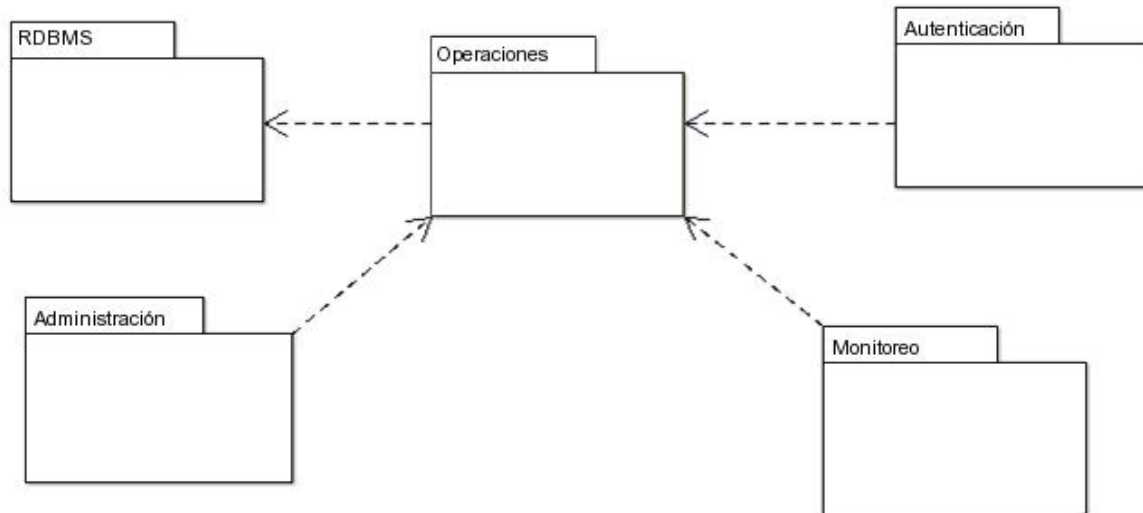


Figura 25. Diagrama de paquetes del sistema

Con el diagrama de paquetes (Figura 39) se muestran las dependencias entre clases. Los diagramas de clases expresan las dependencias entre clases cuando estas pertenecen a paquetes diferentes.

En este caso, tal y como se muestra en el diagrama es la dependencia, pero en nuestro sistema las únicas clases son las del paquete RDBMS, que anteriormente se explico con mayor precisión.

Seguridad en los scripts

La parte de seguridad es fundamental en este sistema, aquí consideramos lo que viene siendo autenticación para que puedan acceder al sistema y para darle mayor seguridad estamos utilizando cifrado y detalles como autenticación para que no sea posible la ejecución de los archivos en forma individual para burlar la seguridad del sistema.

Para darle mayor seguridad, el sistema autentica en todas las páginas, sino esta siendo ejecutada en al página principal entonces es redireccionada a la página principal y no permite ejecución de código.

También se considero utilizar el método post en el envío de información del sistema para que el usuario no tenga a la vista todos los datos que se están transfiriendo.

El cifrado o "encryption" es la traducción de datos a un código secreto. El cifrado es la manera más eficaz de proteger datos, ya que para leer los archivos cifrados es necesario tener acceso a una clave secreta o contraseña que permita descifrar la información. Los datos no cifrados se denominan texto sin formato, mientras que los cifrados se conocen como texto cifrado.

Existen dos tipos principales de cifrado: cifrado asimétrico (también llamado cifrado por clave pública) y cifrado simétrico.

4.2. Instalación y pruebas del sistema

El sistema esta diseñado para poder ser instalado en ambientes Windows y Unix, sin embargo para la instalación y pruebas fueron usados los siguientes elementos:

Hardware

- PC armada, tarjeta madre Intel D915GVWB
- 1 GB en RAM
- 80 GB disco duro

Software

- Sistema operativo Fedora 3.0
- Servidor Web Apache versión 2.2.2
- Servidor de bases de datos Postgresql 8.1
- Modulo PHP versión 5

Prueba unitaria y de integración

La prueba unitaria consiste en probar los componentes individualmente de un software. En la prueba unitaria, se hicieron pruebas a cada componente del sistema, en cada uno de ellos se hizo una prueba de autenticación, otra prueba a cada una de las operaciones que debe realizar cada componente de nuestro sistema (como altas, bajas, modificación de datos y listado correcto de datos).

La prueba de integración consiste en probar varios componentes juntos. El objetivo primario es encontrar defectos en las interfaces entre componentes.

En la prueba de integración cada parte del sistema se integro a su subsistema. Y se realizaron pruebas de acoplamiento con cada subsistema. Como en el subsistema de seguridad la parte de servidores se probó para que no generara ningún error en nuestro sistema.

Prueba de sistema

Para este sistema se realizaron las siguientes pruebas:

La prueba de seguridad se realizó cuando se hizo la autenticación para poder acceder al sistema aunado con la prueba de querer ejecutar cualquier programa en forma individual para burlar la autenticación, y para estas dos partes el sistema respondió en forma satisfactoria.

En la prueba de instalación también dio un buen resultado; ya que para poder ser instalado en cualquier servidor lo único que necesita es cargar los archivos en el servidor Web, ejecutar el script para crear la base de datos Administrator, y al archivo principal indicarle el nombre del servidor, el RDBMS que manejara, el login y la contraseña para que el sistema pueda entrar a la base de datos Administrator.

Debido a que es un sistema que puede manejar varios RDBMS, se configuro Administrator para trabajar con PostgreSQL, MySQL y Sybase, pero debido a los múltiples RDBMS no dio tiempo de hacer pruebas con todos. Y por la falta de recursos y tiempo para entrega del modulo no se hicieron pruebas en otras redes.

Prueba de aceptación

Es la prueba previa a poner en producción el software. El objetivo es verificar que el software está listo y puede ser usado por el usuario final para realizar las funciones y tareas para las que fue construido. Dado que se trata de un sistema complicado se considera que es necesario realizar mas pruebas para poder ser aceptado, sobre todo en la parte de seguridad que es en donde podría ser mas propenso el sistema a ser dañado debido a que se maneja a nivel de Web.



5 CONCLUSIONES

Conclusiones

Cumplimiento de requerimientos

En cuanto a los requerimientos concluimos que todos fueron satisfechos satisfactoriamente; ya que nuestro sistema crea bases de datos en los RDBMS utilizados por el sistema, se puede controlar el acceso al sistema con el uso del login y la contraseña, y se monitorea el tamaño de esta y las bases de datos pueden ser administradas desde cualquier parte gracias a la Internet. El único requerimiento que no pudimos concluir fue el del comportamiento, porque no encontramos la forma de monitorear el estado del RDBMS con PHP.

Cumplimiento de tiempos

En cuanto a los tiempos no fue posible concluirlos satisfactoriamente; ya que nos tardamos mas por la investigación de comandos RDBMS. Esto es debido a que cada RDBMS maneja diferentes comandos para algunas peticiones.

Dificultad de desarrollo del proyecto

El objetivo de nuestro trabajo de tesis era desarrollar un modulo dinámico de administración de RDBMS con software libre. El modulo dinámico de administración de bases de datos fue un sistema difícil de desarrollar por lo poco común que son este tipo de aplicaciones.

La dificultad del sistema fue debido a que cada RDBMS maneja variantes en cuanto a tablas de sistema, y otras diferencias en cuanto al SQL estándar (aunque existe un estándar), se torna difícil poder trabajar con muchos de ellos en un solo sistema. La fase del desarrollo es complicada debido a lo mismo.

Pero gracias a la ingeniería de software y a UML, se pudo hacer un mejor desempeño en cuanto a planeación del sistema y estructuración.

La satisfacción de usuarios

Nuestro sistema lo planeamos desde el principio para que funcionara de la manera más inteligente posible, esta desarrollado en Web para que pueda ser administrado si se requiere desde cualquier parte del mundo, y esta enfocado solo a la creación de base de datos, para que el usuario pueda utilizar las bases de datos para páginas Web o para aplicaciones Java, Visual.Net. etc. Y esta hecho con software libre para que el usuario no pague por las licencias de estos productos.

En cuanto al sistema es ahora un prototipo, es decir, esta resuelto el principio de interacción para que el usuario pueda crear cualquier base de datos en cualquier RDBMS y desde cualquier servidor, y monitorea el tamaño de esta,

junto con unos niveles de seguridad que vienen integrados. Para que sea del completo agrado de administradores de base de datos, se requiere agregar otras funciones.

Las perspectivas a Futuro

Esperamos que este proyecto sea mejorado por nuevos estudiantes; ya que nosotros sabemos que todo producto puede ser mejorado, considerando además que constantemente nacen nuevas herramientas y lenguajes más potentes, y que al final de cuentas todo lo que hacemos es para que el hombre pueda trabajar más rápido, con el mínimo esfuerzo y tiempo.

Consideramos también que el desarrollo de esta tesis nos servirá para poder desarrollar aplicaciones Web más potentes debido a que resolvimos el problema de generalizar la creación de base de datos desde cualquier RDBMS soportado por PHP y en cualquier servidor.



6 GLOSARIO

Glosario

API (Application Program Interface)

Conjunto de especificaciones usadas para establecer comunicación entre componentes de software. Representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas interiores y los superiores del software.

ASP.NET

Es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web domésticos, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

BASE DE DATOS

Conjunto de archivos interrelacionados que es creado y operado por un sistema de administración de bases de datos (DBMS), se trata de cualquier conjunto de datos almacenado electrónicamente.

CERN (Conseil Europeen pour la Recherche Nucleaire)

Institución europea, creada en 1954, con sede en Ginebra, que desarrolló para sus necesidades internas el primer navegador y el primer servidor Web.

CICLO DE DESARROLLO DE UN SISTEMA

Secuencia de eventos en el desarrollo de un sistema de información en los que se involucran tanto el usuario como el programador. La secuencia es 1) Análisis y diseño del sistema, 2) Aprobación del usuario, 3) Programación, 4) Implementación.

CGI (Common Gateway Interface)

Programas usados para hacer llamadas a rutinas o controlar otros programas o bases de datos desde una página Web. Un script CGI es una pequeña aplicación que se ejecuta en el servidor, normalmente a petición de una página Web alojada en el mismo. Los CGI proporcionan la posibilidad de acceder a bases de datos, intercambio de información a través de formularios HTML, administración de claves de accesos, utilidades de búsqueda, etc. Los script CGI pueden elaborarse en cualquier lenguaje, aunque el más usado es PERL y VB.

CRIPTOGRAFÍA (del griego *kryptos*, "ocultar", y *grafos*, "escribir", literalmente "escritura oculta")

Es el arte o ciencia de cifrar y descifrar información utilizando técnicas matemáticas que hagan posible el intercambio de mensajes de manera que sólo puedan ser leídos por las personas a quienes van dirigidos.

COOKIE

Una cookie es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en

posteriores visitas. Las inventó Lou Montulli, un antiguo empleado de Netscape Communications. Al ser el protocolo HTTP incapaz de mantener información por sí mismo, para que se pueda conservar información entre una página vista y otra (como *login* de usuario, preferencias de colores, etc), ésta debe ser almacenada, ya sea en la URL de la página, en el propio servidor, o en una *cookie* en el ordenador del visitante.

COLDFUSION

Es una herramienta que corre en forma concurrente con la mayoría de los servidores Web de Windows, Linux y Solaris (también en servidores web personales en Windows 98 y puede ser usado para intranets). El servidor de aplicaciones web de ColdFusion trabaja con el servidor HTTP para procesar peticiones de páginas Web e interactúa de manera simple con bases de datos

CONTRASEÑA SEGURA

Es una contraseña que otras personas no pueden determinar fácilmente adivinándola o utilizando programas automáticos. Tener una contraseña segura es importante para proteger la información personal.

CONTROL ACTIVEX

Es un componente de software que se puede descargar y que se utiliza para ampliar las funciones de Internet Explorer a través de elementos de la interfaz de usuario, como menús emergentes, botones, etcétera.

CSS (Cascade Style Sheet)

Es un término conocido en programación y consiste en un conjunto de instrucciones HTML que definen elementos de diseño generales de un grupo de páginas Web y se suele usar para ahorrar trabajo y para que se reconozca rápidamente que las páginas Web que forman el conjunto pertenecen todas al mismo tema u organización.

DBA(Database administrator)

Administrador de bases de datos. Individuo responsable del mantenimiento físico y lógico de una base de datos.

DBMS(Data Base Management System)

Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos de una base de datos.

DDL (Data Definition Language)

Lenguaje de definición de datos, que permite crear definiciones de como deben estar organizados los datos en una base de datos. Este lenguaje permite crear, modificar y eliminar tablas de una base de datos.

DHTML

Sigla de Dynamic HTML o HTML dinámico, tecnología que permite dotar a las páginas Web de mayor interactividad a través de efectos especiales y

animaciones entre otros medios. DHTML integra las tecnologías: HTML tradicional, CSS, DOM y scripts.

FIREWALL

Elemento físico ó lógico en una red cuyo fin es prevenir el acceso no autorizado a recursos ó información.

FLASH

Flash es la tecnología más comúnmente utilizada en el Web que permite la creación de animaciones vectoriales. El interés en el uso de gráficos vectoriales es que éstos permiten llevar a cabo animaciones de poco peso, es decir, que tardan poco tiempo en ser cargadas por el navegador

HIPERVÍNCULOS

Secciones especiales en los documentos con las que los usuarios pueden interactuar (típicamente pulsando sobre ellos con el teclado o el ratón) para ir saltando de unos documentos a otros.

HTML (HyperText Markup Language)

Lenguaje simple de etiquetas usado para crear documentos de hipertexto que pueden ser trasladados de una plataforma a otra. Los documentos HTML son documentos SGML con una semántica genérica que es apropiada para representar información de un amplio rango de aplicaciones.

HTTP (*HyperText Transfer Protocol*)

Protocolo utilizado para el intercambio de los archivos de hipertexto (escritos en HTML) a través de Internet, entre un software cliente por una parte y un servidor en el otro, operando ambos bajo este protocolo. El hipertexto es el contenido de las páginas Web, y el protocolo de transferencia es el sistema mediante el cual se envían peticiones las peticiones de acceder a una página Web, y la respuesta de esa Web remitiendo la información que se verá en pantalla.

INGENIERIA DE SOFTWARE

Es la rama de la ingeniería que crea y mantiene las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos.

IIS (Internet Information Server)

Esta solución de software, conocida por las siglas IIS engloba una serie de herramientas que permiten controlar sitios Web, FTP, SMTP (correo saliente) y Servicio de noticias. Además, ofrece el soporte necesario para crear páginas dinámicas (ASP), y aplicaciones para Internet que permiten la conexión y acceso a bases de datos.

INTERFAZ

Conexión o interacción entre hardware, software y usuario. Como interfaces de hardware se tiene a los medios físicos que transportan las señales eléctricas en

orden definido. Como interfaces de software tenemos la estructura que usan los programas para comunicarse entre si. Como interfaces de usuario tenemos a aquellos medios por los cuales el usuario accede o se comunica con una computadora.

INTERNET EXPLORER

Es un navegador de Internet producido por Microsoft para su plataforma Windows y más tarde para Apple Macintosh. Fue creado en 1995 tras la adquisición por parte de Microsoft del código fuente de Mosaic, un navegador desarrollado por Spyglass, siendo rebautizado entonces como Internet Explorer.

INTRANET

Red informática de comunicación privada de una empresa u organización de cualquier tipo que funciona con las mismas tecnologías utilizadas en la red Internet y que, por lo tanto, puede conectarse a ella de forma sencilla dada su misma naturaleza tecnológica.

IP (INTERNET PROTOCOL)

Protocolo estándar utilizado en Internet para la comunicación entre dispositivos. Actualmente existen 2 versiones de este protocolo, el Ipv4 y el Ipv6. También se utiliza el término IP como simplificación de dirección IP.

ISAPI (Internet Server API)

Es una API para el servidor Web IIS (Internet Information Server) de Microsoft. La ISAPI le permite a los programadores desarrollar aplicaciones basadas en Web que corran mucho más velozmente que los programas CGI convencionales porque están más íntimamente integrados con el servidor Web. Además del IIS, varios servidores Web de otros fabricantes soportan la ISAPI.

ISO (International Standardization Organization)

Entidad gestionada por la ONU que se encarga de definir los estándares de todo tipo que deben seguir sus países miembros.

JAVA

Lenguaje de programación desarrollado en 1991 en la firma Sun Microsystems por James Gosling cuya principal característica es la de poseer un código fuente compatible con todas las principales plataformas informáticas del mercado.

JAVASCRIPT

Lenguaje de programación parecido al Java, pero con la diferencia de que éste incorpora los programas dentro del fichero HTML.

MODELO

Representación real de los datos obtenidos a partir de la información disponible. Ejemplos son el modelo de estructura y el modelo de estilo de que representan la estructura analítica y la información de estilo asociada a un documento. El modelo podría ser un árbol, o un grafo orientado, o cualquier otra.

NSAPI (Netscape Server Application Programming Interface)

Interfaz de Programación de Aplicaciones para Servidores Netscape. Especificaciones de programación de servidores Internet de Netscape para aplicaciones back end.

Página de Servidor Activo (ASP)

Tipo especial de página HTML la cual contiene unos pequeños programas (también llamados scripts) los cuales son ejecutados en servidores Microsoft Internet Information Server antes de ser enviados al usuario para su visualización en forma de página HTML. Habitualmente esos programas realizan consultas a bases de datos y los resultados de esas consultas determinan la información que se envía a cada usuario específico. Los archivos de este tipo llevan el sufijo .asp.

Página de Servidor Java (JSP)

Tipo especial de página HTML que contiene unos pequeños programas (también llamados scripts) que son ejecutados en servidores Netscape antes de ser enviados al usuario para su visualización en forma de página HTML. Habitualmente esos programas realizan consultas a bases de datos y los resultados de esas consultas determinan la información que se envía a cada usuario específico. Los archivos de este tipo llevan el sufijo .jsp.

Página Inicial (Homepage)

Es la página Web de entrada a un lugar del WWW la cual es considerada la página principal.

Página Personal

Archivo que contiene la información sobre una persona almacenada de forma individualizada en un servidor Internet, habitualmente WWW.

Página Web

Resultado en hipertexto o hipermedia que proporciona un navegador del WWW después de obtener la información solicitada. Su contenido puede ir desde un texto corto a un voluminoso conjunto de textos, gráficos estáticos o en movimiento, sonido, etc. Algunas veces el citado término es utilizado incorrectamente en orden de designar el contenido global de un sitio Web, cuando en ese caso debería decirse "sitio Web".

PERL(PracticalExtraction and Report Language)

Lenguaje de programación utilizado en el WWW a través de un CGI, principalmente para realizar consultas a bases de datos como Oracle, SQL-Server, SyBase, etc, o a herramientas locales como WAIS. Perl es un lenguaje para manipular textos, archivos y procesos, proporciona una forma fácil y legible para realizar trabajos que normalmente se realizarían en C o en un shell. Perl nació y se ha difundido bajo el sistema operativo UNIX, aunque existe para otras plataformas. Perl fue desarrollado por Larry Wall, y está distribuido libremente bajo la filosofía de la GNU.

PHP(PHP Hypertext Preprocessor)

Lenguaje de programación tipo script e interpretador de disponibilidad libre, ofrecido bajo licenciamiento de Fuente Abierta y utilizado primordialmente en servidores Web bajo Linux. PHP, originalmente conocido como Personal Home Page Tools (herramientas para páginas principales), ahora significa PHP Hypertext Preprocessor. PHP es una alternativa a la tecnología ASP de Microsoft. Al igual que el ASP, el código de PHP está dentro de una página Web en conjunto con código HTML. Antes de que la página sea enviada al usuario que la ha solicitado, el servidor Web solicita al PHP que interprete y ejecute las instrucciones del código PHP. Estas instrucciones pueden ser de cualquier tipo, como por ejemplo, una consulta a una base de datos y la presentación de su resultado.

PROTOTIPO

Modelo mediante el cual se pueden obtener los requisitos para la construcción de un sistema de forma efectiva.

PORTAL

Sitio Web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, compra electrónica, etc.

SERVIDOR

Computadora de alta disponibilidad conectada a Internet que presta servicio de hospedaje de datos.

SISTEMA

Conjunto de componentes relacionados que interactúan para lograr un objetivo.

SISTEMA MONOUSUARIO

Sistema en el que solo puede ejecutarse un proceso a la vez, cuenta con el control absoluto de los recursos de la computadora.

SISTEMA MULTIPROCESO

Sistema en el que más de dos aplicaciones se ejecutan al mismo tiempo por una sola computadora.

SISTEMA MULTIUSUARIO

Sistema de computación con varias terminales, habilitadas para que varios usuarios, cada uno con su propia terminal, usen la misma computadora.

SGML (Standard Generalized Markup Language)

Lenguaje de Marcado Generalizado Estandarizado. Estándar para lenguajes de edición, particularmente aquellos que se usan para el intercambio, manejo y publicación electrónica de documentos.

SPAM

Es el correo electrónico no solicitado. Para el remitente es un correo masivo, la mayoría de las veces obtenido a través de un spambot (robot electrónico) o compañías creadoras de listas de distribución por email con fines publicitarios. Para el destinatario es un correo chatarra que invade su privacidad y además le hace gastar recursos. El spam se ha convertido en uno de los grandes problemas de Internet. Spam es una palabra inglesa que hace referencia a una conserva cárnica: el spiced ham, literalmente “jamón con especias”. Al no necesitar refrigeración fue muy utilizada en todo el mundo, sobre todo por el ejército estadounidense, que ayudo mucho en su difusión. Debido a esto (y a su baja calidad) se ha utilizado este término para hacer referencia a todos los mensajes basura o postings electrónicos de publicidad no deseados que se reciben tanto en los grupos de noticias como en los buzones particulares.

WWW(World Wide Web)

Sistema de hipertexto que funciona sobre Internet mediante un conjunto de servidores http.

XHTML (EXtended HTML)

HTML extendido. Reformulación del HTML 4.0 basada en una aplicación del XML 1.0.

XML (EXtensible Markup Language)

Lenguaje de Marcado Extensible. XML es un subconjunto de SGML desarrollado con el fin de hacer compatible el propio SGML con la red.



7 BIBLIOGRAFÍA

Bibliografía

- Lawrence Pfleeger Shari, “Ingeniería de software Teoría y Práctica” Pearson Educación, 2002
- G. Booch, J Rumbaugh y I. Jacobson, “El lenguaje Unificado de Modelado, Addison Wesley, 1999”
- R.S. Presuman. “Ingeniería del Software. Un enfoque práctico” McGraw Hill, 5ª Edición, 2002.
- Robert T. Grauer y Maryann Barber, “Introducción a la Informática con Microsoft Office XP y 2000” Pearson Prentice Hall Primera edición
- Enrique de Alarcón Álvarez, “Diccionario de Informática e Internet. Guía práctica para usuarios” Editorial Anaya Multimedia
- Enrique de Alarcón Álvarez. “Diccionario de términos informáticos e Internet” Edición 2005 Editorial Anaya multimedia

Breve historia de la programación web

http://www.ciberaula.com/curso/aspnet/que_es/, consultada 10 de agosto de 2004

Páginas web estáticas vs páginas web dinámicas. Las bases de datos en Internet

http://mosaic.uoc.edu/old/2_opinion2/modeljuny/carlos_casado.html, consultada 7 de septiembre de 2004

Conceptos bases de datos

<http://www.monografias.com/trabajos26/seguridad-base-datos/seguridad-base-datos.shtml#concep>, consultada 2 de enero de 2005

El modelo relacional de bases de datos

<http://www.inegi.gob.mx/inegi/contenidos/espanol/prensa/Contenidos/Articulos/tecnologia/relacional.pdf>, consultada el 20 de febrero de 2005

Modelo_er

http://www.udistrital.edu.co/comunidad/profesores/rfranco/modelo_er.htm, consultada el 11 de marzo de 2005

Introducción a los conceptos de bases de datos

Manejador de Bases de Datos

<http://www.ljvcsystems.com.ar/pages/BaseDatos.htm>, consultada el 23 de marzo de 2005

Tipos de bases de datos

http://www.wikilearning.com/tipos_de_bases_de_datos-wkccp-14863-3.htm, consultada el 3 de abril de 2005

Definiciones formales y arquitecturas

Bases de datos

<http://elies.rediris.es/elies9/4-1-3.htm>, consultada el 22 de abril de 2006

Diccionario estadístico

SQL

<http://www.estadistico.com/dic.html?p=4222&PHPSESSID=9f12df39e8ab4baf1a32bf4caa441b9>, consultada el 9 de junio de 2005

Estándares ANSI SQL

<http://sunsite.unam.mx/servidores/docs/bd/ANSI%20SQL.pdf>, consultada el 1 de agosto de 2005

Paradigmas de Modelado de Base de Datos I. CI5311.

<http://www.bd.cesma.usb.ve/ci5311/sd05/apuntesParadigmas.pdf>, consultada 35 de octubre de 2005

1.1 Historia y funciones del DBA

<http://sistemas.itlp.edu.mx/tutoriales/admonbasedat/uni1tem1.htm>, consultada 8 de diciembre de 2005

3. Tipos de bases de datos

http://www.wikilearning.com/tipos_de_bases_de_datos-wkccp-14863-3.htm, consultada el 17 de enero de 2006

Aplicaciones Multinivel

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html>, consultada el 29 de enero de 2006

Servidores Web

<http://www.masadelante.com/faq-servidor-web.htm>, consultada 9 de octubre de 2005

Manual de PHP

http://www1.us.es/pautadatos/publico/personal/pas/2166/1498/php_manual.pdf, consultada el 29 de octubre de 2005

Tecnología de edición de paginas Web

<http://www.monografias.com/trabajos11/pawe/pawe.shtml>, consultada el 5 de enero de 2006

Java 2 Enterprise Edition - Enciclopedia

http://enciclopedia.us.es/index.php/Java_2_Enterprise_Edition, consultada el 27 de enero de 2006

Cold Fusion

Características y Funciones

<http://www.uca.edu.sv/investigacion/bdweb/reportes/coldfusion.html>, consultada el 3 de febrero de 2006

Ingeniería de software

http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software, consultada el 19 de mayo de 2006

El Ciclo de vida

<http://www.getec.etsit.upm.es/docencia/gproyectos/planificacion/cvida.htm>

Diccionario

<http://www.seguridad.unam.mx/usuario-casero/main.dsc?dicc=5257>, consultada el 10 de enero de 2007

Que es Flash

<http://www.desarrolloweb.com/articulos/1067.php>, consultada el 10 de enero de 2007

Wikipedia

<http://es.wikipedia.org/wiki/HTML>, consultada el 10 de enero de 2007

Glosario

http://www.marcelopedra.com.ar/glosario_1.htm, consultada el 10 de enero de 2007