



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA DE BASES DE DATOS PARA LAS
REDES DE VOZ Y DATOS DE LA UNAM**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

CORAL ALEJANDRA MONTES CARMONA

DIRECTOR DE TESIS:

ING. LUCILA PATRICIA ARELLANO MENDOZA



CIUDAD UNIVERSITARIA

MÉXICO, D.F., 2011.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

*Dedicado a las personas que más amo y
las más importantes de mi vida:*

Juan Montes Hernández

Alejandra Carmona Silva

Jessica Crystal Montes Carmona

Aarón Martínez Álvarez

Coral Alejandra Montes Carmona

AGRADECIMIENTOS

A DIOS:

Gracias mi amado Dios por la vida que me das y porque siempre estás conmigo, por darme siempre lo mejor, por cada una de las cosas que he vivido, que tengo y que he logrado, porque en todo momento me has cuidado, porque de ti proviene la inteligencia y la sabiduría, porque me das salud, fortaleza, paz y tu amor perfecto e incomparable, hermoso e infinito. Gracias por ser mi mejor amigo y todo lo que necesito, por escucharme, por bendecirme, por demostrarme tu amor siempre. Eres lo más valioso y lo más precioso en mi vida. Vivo amándote cada día más y agradeciéndote por la vida, la salud, la compañía y el amor de mis seres amados, por todas las cosas que me das y que me permites realizar como el alcanzar esta meta que significa tanto para mí. TE AMO MI DIOS.

A MIS PADRES:

Gracias por darme un hogar lleno de amor, de respeto, armonía y paz, donde la unidad es muy importante. Gracias por enseñarme lo que no se aprende en la escuela y que es más valioso que todo ello, por los valores y principios que me inculcaron.

Gracias a ustedes he podido llegar hasta aquí y no hay palabras que puedan expresar todo mi agradecimiento y toda la admiración que tengo por los dos. Cada día han sido mi mayor fuente de inspiración. Gracias por confiar en mí. Los quiero mucho.

Papá:

Siempre has sido, eres y serás mi mejor ejemplo a seguir. Tu excelencia y responsabilidad en tu trabajo y en la casa me han motivado a dar lo mejor de mí en cada cosa que he realizado y así será siempre. La constancia y perseverancia por conseguir tus objetivos y realizar tus metas no sólo me han llenado de gran satisfacción sino que también alimentan mis ideales y mi esfuerzo para hacer lo mismo en cada reto que se ha presentado a lo largo de mi vida. Gracias por ser el mejor papá del mundo, por darme todo lo que necesité, por tu apoyo, por estar siempre pendiente de mí en todos los aspectos, por tu protección y cuidados, por tus consejos en esas pláticas que guardo en mi mente y en mi corazón, por tu ternura y tu amor. Te admiro mucho por ser un hombre triunfador, inteligente, íntegro, noble y de buen corazón pero sobre todo por ser mi papá. Te quiero muchísimo!!!

Mamá:

Eres una gran mujer, la más fuerte y la más recta que existe. Mi mayor ejemplo de fortaleza y de entrega a Dios y a su familia. Cada día me enseñaste que el mejor camino es el de Dios, con palabras y con hechos. Gracias porque en todo momento me procuraste y nunca olvidaré todo lo que hacías conmigo y por mí y sigues haciendo, siempre me apoyaste y tu determinación me impulsó y me motivó siempre. Gracias por todo tu tiempo y tus

cuidados, por tu esmero y dedicación a mi persona, por preocuparte siempre en que terminara mis proyectos y lo hiciera dando lo mejor de mí, por tus oraciones, por tus palabras y regaños, por tus consejos, por tu amor insuperable de madre y sobre todo por ser mi mejor amiga. Dios supo darme a la mejor mamá. Te admiro y te quiero muchísimo!!!

A mi hermanita Crys:

No sabes cuánto le agradezco a Dios por tu vida, por haberme permitido tener a alguien con quien compartir todo, es la mejor experiencia que he tenido desde que naciste. Eres una súper hermana, mi amiga especial y de ti he aprendido mucho. Quiero decirte que sin ti mi vida no sería igual, tú le das un significado diferente a todo lo que hago ya que es un compromiso que también tengo contigo. Gracias por estar siempre conmigo y apoyarme en todo, por tus palabras, por tus abrazos y besos. Eres muy cariñosa y linda y te valoro mucho. Te quiero demasiado bebé!!!

Aarón Martínez Álvarez:

Gracias mi amor por todo tu apoyo y comprensión, por estar conmigo siempre pero sobre todo gracias por tu amor, ese gran amor que me das y me demuestras cada día y en todo momento, por tu confianza y por ser tan lindo conmigo siempre. Te admiro muchísimo en todos los aspectos y áreas de tu vida. Eres un gran hombre y un gran ejemplo para mí por ser muy trabajador y comprometido en todas las cosas que realizas. Tu amor por Dios me ha impactado como no te imaginas y llegaste a mi vida en el momento preciso. Eres el regalo más hermoso que Dios me ha dado. Eres único, increíble y perfecto para mí. TE AMO MUCHÍSIMO!!!

A mis abuelitos:

Francisca Silva, Francisca Hernández, Eusebio Carmona y Benjamín Montes

Gracias por su amor y por darle vida a mis padres, por esa familia tan grande y hermosa que me han dado, siempre los recuerdo con cariño, admiración y respeto. A ti abuelito Benjamín que puedo disfrutar de tu compañía, gracias por estar siempre presente en los momentos más importantes de mi vida. Te quiero mucho.

A mis tíos:

Isabel, Remedios, Rubén, Alfredo, Arnulfo, Nati, Tomás, Laura, Mere, Honorio, Eusebio, Gregorio, Lola, Isaac, Catalina y Alfredo.

Gracias porque contribuyeron con su ejemplo, con sus experiencias y por brindarme su apoyo en cualquier cosa que requería y que estaba a su alcance. De igual forma gracias por sus palabras, consejos y por llevarme en sus oraciones. Los quiero mucho.

Rosa Isela Hernández Díaz:

Gracias por ser mi mejor amiga, por tus palabras, tus consejos y tu compañía. Todo lo que vivimos en la facultad lo guardo en mi corazón y cada vez que lo recuerdo se dibuja en mi cara una enorme sonrisa. Siempre has sido un gran apoyo para mí y nuestra amistad es muy valiosa y estoy segura que durará por siempre. Te quiero mucho amigaaa!

Daniel Nava Nava:

Gracias por tu apoyo, tu tiempo, por tu disposición y por brindarme tus conocimientos y tu amistad.

Alejandro Cruz Santos:

Gracias por la oportunidad que me brindaste para emprender este proyecto del NIC-UNAM, por apoyarme y depositar tu confianza en mí, eres un gran amigo que siempre me impulsó y a quien le tengo un gran cariño y admiración.

A mis amigos:

Ivonne, Susana, Selene, Celene, Janet, Zamira, Marisol, Ana, Eduardo, Christian, Luis, Juan José, Magno, Mario, David, Paul, José, Mauricio, Roberto, Guillermo, Humberto y Oswaldo.

Gracias por compartir tantos momentos juntos que son inolvidables y muy valiosos para mí. Aprendí mucho con ustedes y de ustedes. Los quiero.

A mi Universidad, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO:

Gracias por albergarme en tus aulas durante todo este tiempo, y ser como mi segunda casa, por darme a los mejores profesores y la educación de la más alta calidad en este país.

Ing. Lucila Patricia Arellano Mendoza:

Gracias por ser mi directora de tesis, por su comprensión, apoyo, disposición, tiempo, paciencia y conocimientos.

A mis profesores:

Gracias porque sin lugar a dudas ustedes formaron los cimientos y me dotaron de los mejores instrumentos para que yo pudiera ir construyendo poco a poco esta realidad y por su valioso ejemplo.

A mis amados pastores que siempre me llevan en sus oraciones, a mis familiares, amigos, compañeros, a mis sinodales y a todos aquellos que de alguna u otra forma han estado presentes en mi vida y que han sido pieza clave y han contribuido para lograr este triunfo.

ÍNDICE

Introducción	1
Antecedentes	9
CAPÍTULO 1. CONCEPTOS BÁSICOS	22
1.1 BASES DE DATOS	22
1.1.1 Cliente – servidor	26
1.1.2 Niveles de abstracción de datos	27
1.1.3 Lenguajes de bases de datos	28
1.1.4 SQL	29
1.1.5 Usuarios y administradores de las bases de datos	36
1.1.6 Interfaces del SGBD	38
1.1.7 Estructura de un sistema de bases de datos	42
1.1.8 Modelos de datos	43
1.2 REDES DE COMPUTADORAS	50
1.2.1 Tipos de redes	50
1.2.2 Arquitectura de protocolos	52
1.2.3 Direcciones IP	58
1.2.4 DNS	62
1.2.5 Voz sobre IP	66
1.3 SISTEMAS OPERATIVOS	67
1.3.1 Funciones básicas de un sistema operativo	67
1.3.2 Componentes de un sistema operativo	69
1.3.3 Clasificación de los sistemas operativos	71
1.3.4 UNIX	74
1.3.5 LINUX	77
1.4 LENGUAJES DE PROGRAMACIÓN	78
1.4.1 Lenguajes de marcas	79
1.4.2 JAVA	85
1.4.3 JAVASCRIPT	87
1.4.4 AJAX	88
1.4.5 PHP.	89
1.5 INGENIERÍA DE PROGRAMACIÓN	91
1.5.1 Metodologías de desarrollo de software	91

CAPÍTULO 2. ANÁLISIS DEL SISTEMA	94
2.1 DEFINICIÓN DEL PROBLEMA	94
2.2 ANÁLISIS DE REQUERIMIENTOS	95
2.3 ALCANCES	97
2.4 CASOS DE USO	98
2.4.1 Requerimientos funcionales del Sistema de Bases de Datos	100
 CAPÍTULO 3. PROPUESTA DE SOLUCIÓN	 121
3.1 EVALUACIÓN Y SELECCIÓN DE HERRAMIENTAS	121
3.2 METODOLOGÍA	123
 CAPÍTULO 4. DESARROLLO DEL SISTEMA	 131
4.1 DISEÑO Y CREACIÓN DEL MODELO	131
4.2 DESARROLLO DE SOFTWARE	149
4.3 IMPLEMENTACIÓN	151
4.4 PRUEBAS	151
4.5 MANTENIMIENTO	153
 Conclusiones	 154
Anexos	157
A. Manual de Usuario	158
B. Glosario	178
Bibliografía	182
Mesografía	183

INTRODUCCIÓN

Los sistemas de información existen desde las primeras civilizaciones. Los datos se recopilaban, se estructuraban, se centralizaban y se almacenaban convenientemente. El objetivo inmediato de este proceso era poder recuperar estos mismos datos u otros datos derivados de ellos en cualquier momento, sin necesidad de volverlos a recopilar, paso que solía ser el más costoso o incluso irrepentible. El objetivo ulterior de un sistema de información, no obstante, era proporcionar a los usuarios información fidedigna sobre el dominio que representaban, con el objetivo de tomar decisiones y realizar acciones más pertinentes que las que se realizarían sin dicha información.

Llamamos base de datos justamente a esta colección de datos recopilados y estructurados que existe durante un periodo de tiempo.

Generalmente, un sistema de información consta de una o más bases de datos, junto con los medios para almacenarlas y gestionarlas, sus usuarios y sus administradores.

Hoy en día, sin embargo, solemos asociar las bases de datos con los ordenadores, y su gestión no suele ser manual, sino altamente automatizada. Más concretamente, la tecnología actual insta a la delegación de la gestión de una base de datos a unos tipos de aplicaciones de software específicas denominadas sistemas de gestión de bases de datos (SGBD) o, simplemente, sistemas de bases de datos.

Un Sistema de Bases de Datos es básicamente un sistema para archivar datos en una computadora, es decir, es un sistema computarizado cuyo propósito general es mantener información y hacer que esté disponible cuando se solicite. La información en cuestión puede ser cualquier cosa que se considere importante para el individuo o la organización a la cual debe servir el sistema.

Las bases de datos son ampliamente usadas. Actualmente el acceso a las bases de datos forma una parte esencial de la vida de casi todas las personas pero debido a que las interfaces de datos ocultan detalles del acceso a las bases de datos, la mayoría de la gente ni siquiera está consciente de que están interactuando con una base de datos.

Las siguientes son algunas de sus aplicaciones más representativas:

- *Banca.* Para información de los clientes, cuentas y préstamos, y transacciones bancarias.
- *Líneas aéreas.* Para reservas e información de planificación. Las líneas aéreas fueron de los primeros en usar las bases de datos de forma distribuida geográficamente (los terminales situados en todo el mundo accedían al sistema de bases de datos centralizado a través de las líneas telefónicas y otras redes de datos).
- *Universidades.* Para información de los estudiantes, matrículas de las asignaturas y cursos.
- *Transacciones de tarjetas de crédito.* Para compras con tarjeta de crédito y generación mensual de extractos.

- *Telecomunicaciones.* Para guardar un registro de las llamadas realizadas, generación mensual de facturas, manteniendo el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.
- *Finanzas.* Para almacenar información sobre grandes empresas, ventas y compras de documentos formales financieros, como bolsa y bonos.
- *Ventas.* Para información de clientes, productos y compras.
- *Producción.* Para la gestión de la cadena de producción y para el seguimiento de la producción de elementos en las factorías, inventarios de elementos en almacenes y pedidos de elementos.
- *Recursos humanos.* Para información sobre los empleados, salarios, impuestos y beneficios, y para la generación de las nóminas.

Como esta lista ilustra, las bases de datos forman una parte esencial de casi todas las empresas actuales. A lo largo de las últimas cuatro décadas del siglo veinte, el uso de las bases de datos creció en todas las empresas.

En los primeros días, muy pocas personas interactuaron directamente con los sistemas de bases de datos, aunque sin darse cuenta interactuaron con bases de datos indirectamente (con los informes impresos como extractos de tarjetas de crédito, o mediante agentes como cajeros de bancos y agentes de reserva de líneas aéreas). Después vinieron los cajeros automáticos y permitieron a los usuarios interactuar con las bases de datos. Las interfaces telefónicas con los computadores (sistemas de respuesta vocal interactiva) también permitieron a los usuarios manejar directamente las bases de datos. Un usuario podía marcar un número y pulsar teclas del teléfono para introducir información o para seleccionar opciones alternativas, para determinar las horas de llegada o salida, por ejemplo, o para matricularse de asignaturas en una universidad.

La revolución de Internet a finales de la década de 1990 aumentó significativamente el acceso directo del usuario a las bases de datos. Las organizaciones convirtieron muchas de sus interfaces telefónicas a las bases de datos en interfaces Web, y pusieron disponibles en línea muchos servicios. Por ejemplo, cuando se accede a una tienda de libros en línea y se busca un libro o una colección de música se está accediendo a datos almacenados en una base de datos. Cuando se solicita un pedido en línea, el pedido se almacena en una base de datos. Cuando se accede a un banco en un sitio Web y se consulta el estado de la cuenta y los movimientos, la información se recupera del sistema de bases de datos del banco. Cuando se accede a un sitio Web, la información personal puede ser recuperada de una base de datos para seleccionar los anuncios que se deberían mostrar. Más aún, los datos sobre los accesos Web pueden ser almacenados en una base de datos.

La importancia de los sistemas de bases de datos se puede juzgar de otra forma: actualmente, los vendedores de sistemas de bases de datos como Oracle están entre las mayores compañías software en el mundo, y los sistemas de bases de datos forman una parte importante de la línea de productos de compañías más diversificadas, como Microsoft e IBM.

Historia

El procesamiento de datos impulsa el crecimiento de los ordenadores, como ocurriera en los primeros días de los computadores comerciales. De hecho, la automatización de las tareas de procesamiento de datos precede a los computadores. El uso de sistemas de bases de datos automatizadas, se desarrolló a partir de la necesidad de almacenar grandes cantidades de datos, para su posterior consulta, producidas por las nuevas industrias que creaban gran cantidad de información. Los predecesores de los sistemas de bases de datos fueron los sistemas de ficheros.

Para obtener los resultados del censo realizado en 1880 de los Estados Unidos de América, tuvieron que pasar alrededor de 7 años. Debido a esta situación, la oficina de censos estadounidense comisionó a Herman Hollerith para que aplicara su experiencia en tarjetas perforadas y llevara a cabo el censo de 1890.

Las tarjetas perforadas fueron inventadas por Herman Hollerith (1860-1929), quien fue denominado el primer ingeniero estadístico de la Historia debido a que en 1884 inventó una computadora llamada “Máquina Automática Perforadora de Tarjetas” con la cual los resultados del censo de 1890 se dieron en 2 años y medio, donde se podía obtener datos importantes como número de nacimientos, población infantil y número de familias. La máquina uso sistemas mecánicos para procesar la información de las tarjetas y para tabular los resultados. Con dicho procesamiento, el censo se terminó en menos tiempo del esperado y como consecuencia de ello, hubo un ahorro de dinero. Así empezó el procesamiento automatizado de datos. Este invento disparó el desarrollo de la tecnología, la industria de los computadores, abriendo así nuevas perspectivas y posibilidades hacia el futuro. Las tarjetas perforadas posteriormente se usaron ampliamente como medio para introducir datos en los computadores.

Las técnicas del almacenamiento de datos han ido evolucionando a lo largo de los años de la siguiente forma:

❖ **Década de 1950 y principios de la década de 1960.**

Se desarrollaron las cintas magnéticas para el almacenamiento de datos. Las tareas de procesamiento de datos tales como las nóminas fueron automatizadas, con los datos almacenados en cintas. El procesamiento de datos consistía en leer datos de una o más cintas y escribir datos en una nueva cinta. Los datos también se podían introducir desde paquetes de tarjetas perforadas e impresos en impresoras.

Las cintas (y los paquetes de tarjetas perforadas) sólo se podían leer secuencialmente, y los tamaños de datos eran mucho mayores que la memoria principal; así, los programas de procesamiento de datos tenían que procesar los datos según un determinado orden, leyendo y mezclando datos de cintas y paquetes de tarjetas perforadas.

Se dice que los sistemas de bases de datos tienen sus raíces en el proyecto estadounidense Apolo de mandar al hombre a la luna, en los años sesenta. En aquella época, no había ningún sistema que permitiera gestionar la inmensa cantidad de

información que requería el proyecto. La primera empresa encargada del proyecto, NAA (North American Aviation), desarrolló un software denominado GUAM (General Update Access Method) que estaba basado en el concepto de que varias piezas pequeñas se unen para formar una pieza más grande, y así sucesivamente hasta que el producto final está ensamblado. Esta estructura, que tiene la forma de un árbol, es lo que se denomina una estructura jerárquica. A mediados de los sesenta, IBM se unió a NAA para desarrollar GUAM en lo que ahora se conoce como IMS (Information Management System). El motivo por el cual IBM restringió IMS al manejo de jerarquías de registros fue el de permitir el uso de dispositivos de almacenamiento serie, más exactamente las cintas magnéticas, ya que era un requisito del mercado por aquella época.

A mitad de los sesenta, se desarrolló IDS (Integrated Data Store), de General Electric. Este trabajo fue dirigido por uno de los pioneros en los sistemas de bases de datos, Charles Bachmann. IDS era un nuevo tipo de sistema de bases de datos conocido como sistema de red, que produjo un gran efecto sobre los sistemas de información de aquella generación. El sistema de red se desarrolló, en parte, para satisfacer la necesidad de representar relaciones entre datos más complejas que las que se podían modelar con los sistemas jerárquicos, y, en parte, para imponer un estándar de bases de datos. Para ayudar a establecer dicho estándar, CODASYL (Conference on Data Systems Languages), formado por representantes del gobierno de EEUU y representantes del mundo empresarial, formaron un grupo denominado DBTG (Data Base Task Group), cuyo objetivo era definir unas especificaciones estándar que permitieran la creación de bases de datos y el manejo de los datos. El DBTG presentó su informe final en 1971 y aunque éste no fue formalmente aceptado por ANSI (American National Standards Institute), muchos sistemas se desarrollaron siguiendo la propuesta del DBTG. Estos sistemas son los que se conocen como sistemas de red, o sistemas CODASYL o DBTG.

❖ **Finales de la década de 1960 y la década de 1970.**

El amplio uso de los discos fijos a finales de la década de 1960 cambió en gran medida el escenario del procesamiento de datos, ya que los discos fijos permitieron el acceso directo a los datos. La ubicación de los datos en disco no era importante, ya que a cualquier posición del disco se podía acceder en sólo decenas de milisegundo. Los datos se liberaron de la tiranía de la secuencialidad. Con los discos pudieron desarrollarse las bases de datos de red y jerárquicas, que permitieron que las estructuras de datos tales como listas y árboles pudieran almacenarse en disco. Los programadores pudieron construir y manipular estas estructuras de datos. Un artículo histórico de Codd en 1970 definió el modelo relacional y formas no procedimentales de consultar los datos en el modelo relacional, y nacieron las bases de datos relacionales. La simplicidad del modelo relacional y la posibilidad de ocultar completamente los detalles de implementación al programador fueron realmente atractivas. Codd obtuvo posteriormente el prestigioso premio Turing de la ACM (Association of Computing Machinery, asociación de maquinaria informática) por su trabajo.

En 1976, Chen presentó el modelo entidad-relación, que es la técnica más utilizada en el diseño de bases de datos. En 1979, Codd intentó subsanar algunas de las deficiencias

de su modelo relacional con una versión extendida denominada RM/T (1979) y más recientemente RM/V2 (1990). Los intentos de proporcionar un modelo de datos que represente al mundo real de un modo más fiel han dado lugar a los modelos de datos semánticos.

❖ **Década de 1980.**

Aunque académicamente interesante, el modelo relacional no se usó inicialmente en la práctica debido a sus inconvenientes por el rendimiento; las bases de datos relacionales no pudieron competir con el rendimiento de las bases de datos de red y jerárquicas existentes. Esta situación cambió con System R, un proyecto innovador en IBM Research que desarrolló técnicas para la construcción de un sistema de bases de datos relacionales eficiente. El prototipo de System R completamente funcional condujo al primer producto de bases de datos relacionales de IBM: SQL/DS. Los primeros sistemas de bases de datos relacionales, como DB2 de IBM, Oracle, Ingres y Rdb de DEC, jugaron un importante papel en el desarrollo de técnicas para el procesamiento eficiente de consultas declarativas. En los principios de la década de 1980 las bases de datos relacionales llegaron a competir con los sistemas de bases de datos jerárquicas y de red incluso en el área de rendimiento. Las bases de datos relacionales fueron tan sencillas de usar que finalmente reemplazaron a las bases de datos jerárquicas y de red; los programadores que usaban estas bases de datos estaban forzados a tratar muchos detalles de implementación de bajo nivel y tenían que codificar sus consultas de forma procedimental. Aún más importante, debían tener presente el rendimiento durante el diseño de sus programas, lo que implicaba un gran esfuerzo. En cambio, en una base de datos relacional, casi todas estas tareas de bajo nivel se realizan automáticamente por la base de datos, liberando al programador en el nivel lógico. Desde su escalada en el dominio en la década de 1980, el modelo relacional ha conseguido el reinado supremo entre todos los modelos de datos. La década de 1980 también fue testigo de una gran investigación en las bases de datos paralelas y distribuidas, así como del trabajo inicial en las bases de datos orientadas a objetos.

❖ **Principios de la década de 1990.**

El lenguaje SQL se diseñó fundamentalmente para las aplicaciones de ayuda a la toma de decisiones, que son intensivas en consultas, mientras que el objetivo principal de las bases de datos en la década de 1980 fue las aplicaciones de procesamiento de transacciones, que son intensivas en actualizaciones. La ayuda a la toma de decisiones y las consultas reemergieron como una importante área de aplicación para las bases de datos. Las herramientas para analizar grandes cantidades de datos experimentaron un gran crecimiento de uso. Muchos vendedores de bases de datos introdujeron productos de bases de datos paralelas en este periodo, así como también comenzaron a ofrecer bases de datos relacionales orientadas a objetos.

❖ **Finales de la década de 1990.**

El principal acontecimiento fue el crecimiento explosivo de World Wide Web. Las bases de datos se implantaron mucho más extensivamente que nunca antes. Los

sistemas de bases de datos tienen ahora soporte para tasas de transacciones muy altas, así como muy alta fiabilidad y disponibilidad 24x7 (disponibilidad 24 horas al día y 7 días a la semana, que significa que no hay tiempos de inactividad debidos a actividades de mantenimiento planificadas). Los sistemas de bases de datos también tuvieron interfaces Web a los datos.

❖ Siglo XXI

Los principios del siglo XXI han sido testigos de la emergencia de XML y su lenguaje asociado, XQuery, como nueva tecnología de las bases de datos. En este periodo también se ha podido presenciar el crecimiento de las técnicas de “informática autónoma / administración automática” para la minimización del esfuerzo de administración. En la actualidad existe una gran cantidad de alternativas en línea que permiten hacer búsquedas orientadas a necesidades específicas de los usuarios, una de las tendencias más amplias son las bases de datos que cumplan con el protocolo Open Archives Initiative – Protocol for Metadata Harvesting (OAI-PMH) los cuales permiten el almacenamiento de gran cantidad de artículos que ofrecen una mayor visibilidad y acceso en el ámbito científico y general.

Como vemos, los Sistemas de Bases de Datos han ido evolucionando de manera creciente debido a la gran demanda que han tenido desde su aparición y el éxito obtenido se debe a que no sólo brindan todo lo necesario para la recuperación y manipulación de la información sino que además ofrecen otros beneficios tales como la posibilidad de descarga de información, envío por correo electrónico, fácil acceso, entre otros, que aunados a esto hacen de dichos sistemas una increíble herramienta en cualquier empresa o lugar que se requieran. Asimismo, uno de los grandes medios de comunicación que ha tomado mayor relevancia en los últimos años, tanto por el rápido crecimiento que ha presentado como por el gran uso que se le da, es la Internet, la cual es una gran fuente de información abierta al público, además de ser un medio de comunicación por excelencia sin importar la distancia, al permitir el acceso a: correo electrónico, conversaciones escritas en línea, transmisión de video, voz, señales de televisión, etc.

Por consiguiente, al combinar un Sistema de Bases de Datos con la Internet, se obtiene un Sistema de bases de datos que se consultará vía web haciendo uso de la conexión a la red mundial y con esto se logra vencer el reto de la ubicación geográfica de las personas y así sólo necesitar de una computadora con acceso a Internet para poder consultar la información que se desee en cualquier momento y lugar.

Es por esta razón que para el desarrollo de la presente tesis se hace uso de lo previamente mencionado, logrando así un Sistema de Bases de Datos para las Redes de Voz y Datos de la UNAM.

OBJETIVO

Proveer a NIC-UNAM de un Sistema de Bases de Datos que cumpla cabalmente con los nuevos requerimientos, contribuyendo de esta forma en el mejor desempeño en la administración por parte de NIC-UNAM y así poder impulsar el desarrollo y mejora en la calidad de las funciones que se realizan y los servicios que se ofrecen. Asimismo, reforzar la seguridad y el control de la información.

DESAROLLO

La tesis fue desarrollada utilizando la metodología RUP debido a que ésta se caracteriza por ser un proceso iterativo e incremental, estar centrado en la arquitectura y dirigido por los casos de uso. Dicha metodología utiliza algunas de las mejores y más modernas prácticas de desarrollo de software que han sido probadas en un gran número de proyectos y organizaciones; además, en el diseño y documentación se utiliza UML (Unified Modeling Language), el cual es ampliamente utilizado en la actualidad. Asimismo incluye artefactos, roles y asigna tareas y responsabilidades (quién hace qué, cuándo y cómo), de forma disciplinada.

El presente trabajo está desarrollado de la siguiente manera:

En la parte de antecedentes se explicará al lector la necesidad e importancia de este Sistema debido a su contenido y a su utilidad, ya que alberga las redes de voz y datos de la UNAM (RedUNAM). Por lo que se describirá la estructura de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (TIC-UNAM) y la ubicación del NIC-UNAM dentro de ella y la manera en que éste se relaciona con los otros departamentos. Asimismo, son presentados los antecedentes respectivos al Sistema que se desarrollará y los motivos por los cuales se determinó crear este Sistema de Bases de Datos.

Posteriormente, se divide en 4 capítulos con el siguiente contenido:

- ❖ En el capítulo 1 se presentan los conceptos básicos de los temas fundamentales que se utilizaron en la realización del Sistema de Bases de Datos, los cuales son: Bases de Datos, Redes de Computadoras, Sistemas Operativos, Lenguajes de Programación e Ingeniería de Programación. Mismos que representan las bases que sustentan el desarrollo del proyecto.
- ❖ En el capítulo 2 se desarrollan las dos primeras fases de la metodología RUP, es decir, la fase Inicio o Concepción y la fase de Elaboración. Logrando así determinar la visión del proyecto y la arquitectura óptima mediante la definición del problema, el análisis de los requerimientos, la definición de los alcances y la construcción de los casos de uso.
- ❖ En el capítulo 3 se presenta la propuesta de solución al problema previamente descrito en el capítulo 2 y las respectivas herramientas a utilizar para poder llevar a cabo dicha propuesta, así como la justificación correspondiente a cada una de ellas.

También, se describe más ampliamente la metodología RUP presentada en el capítulo 1. De igual forma se explica el patrón de diseño de arquitectura de software MVC y CakePHP que es el framework que implementa dicho patrón y está escrito en PHP, el cual es utilizado en el desarrollo de la aplicación web.

- ❖ En el capítulo 4 se desarrollan las dos últimas fases de la metodología RUP: Construcción y Transición, donde se va observando el avance del producto de software mediante el diseño y la creación del modelo, así como el desarrollo de software, con todo lo que esto implica hasta lograr instalar el Sistema y hacer las pruebas correspondientes para dejarlo totalmente funcional.

Al término de los capítulos se presentan las conclusiones que se obtuvieron con el desarrollo de este Sistema.

Cabe mencionar que tanto el glosario como el manual de usuario se encuentran al final como anexos, para facilitar la comprensión de los términos utilizados en cuanto al glosario se refiere y respecto al manual de usuario como parte de la documentación para el usuario del Sistema, en el que se describe éste de manera funcional.

Al final, se presentan tanto la bibliografía como la mesografía consultada por si se requiere ampliación de algún tema en particular y como referencias.

ANTECEDENTES

TIC-UNAM

La Dirección General de Cómputo y de Tecnologías de Información y Comunicación (TIC) de la UNAM (antes Dirección General de Servicios de Cómputo Académico -DGSCA-) es la entidad universitaria encargada de la operación de los sistemas centrales de cómputo académico y de las telecomunicaciones de la institución; su esfuerzo más amplio es la capacitación en tecnologías de la información, de prospección e innovación y de asimilación de estas tecnologías en beneficio de la Universidad y de la sociedad en general.

RedUNAM

La RedUNAM es una compleja Red de Redes, pionera y líder en México, que transporta simultáneamente enormes volúmenes de paquetes de voz, datos y video.

Es una de las redes educativas de telecomunicaciones digitales de mayor densidad y más avanzadas en América Latina. Cumple con la tarea estratégica de intercomunicar a la comunidad universitaria entre sí, y a ésta con el resto de la comunidad académica y científica mundial.

RedUNAM es uno de los centros de tráfico de información más importantes del país. Es un punto imprescindible de conexión a la red mundial Internet: su infraestructura, altamente tecnificada, da servicio de conexión y de enlace a toda la república y al resto del mundo.

Inicio y desarrollo

El final de los años 60s y el principio de la década de los 70s marcaron para la UNAM, la etapa de inicio de las comunicaciones telefónicas y de datos.

Es en ese periodo cuando se realizan las primeras conexiones de teletipos hacia una computadora central, utilizando líneas telefónicas de cobre, de la recién instalada red telefónica dentro de la Institución.

Rápidamente esta tecnología es usada al interior de la UNAM y difundida al exterior, por ello se efectúan una gran cantidad y diversidad de conexiones, de terminales de caracteres, de graficación e impresión, hasta la interconexión de estaciones de trabajo -remotas todas ellas- manejando líneas telefónicas. A partir de la segunda parte de la década de los 80s surge, en la UNAM, la búsqueda de cambios en las comunicaciones.

Así en 1987, la UNAM establece la primera conexión a la Red Académica de C o BITNET, mediante enlaces telefónicos, desde la Ciudad Universitaria hasta el Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), y de ahí hasta San Antonio, Texas en los EUA.

Posteriormente, la UNAM buscó consolidar su enlace a esa red internacional mediante la computadora IBM 4381, la cual sirvió como residencia del correo electrónico y otros servicios de BITNET; dentro de ese proceso, se inició la conexión de terminales IBM con emulación 3270, estableciéndose además, un enlace con la Red TELEPAC de la SCT, bajo la finalidad, nunca lograda, de brindar este servicio a nivel nacional.

No fue sino hasta 1989, cuando la UNAM a través del Instituto de Astronomía establece un convenio de enlace a la red de la NSF en EUA, el cual se realizó haciendo uso del satélite mexicano Morelos II entre el Instituto de Astronomía de la UNAM y el UCAR-NCAR con residencia en Boulder Colorado; además, se llevó a cabo el primer enlace para conectar las redes de área local, entre el Instituto de Astronomía y la Dirección General de Servicios de Cómputo Académico, utilizando enlaces de fibra óptica.

A partir de ese momento se inició, dentro de la UNAM, una revolución en las comunicaciones, así como la adquisición masiva de computadoras personales y su interconexión e intercomunicación en redes de área local, principalmente en las dependencias del Subsistema de la Investigación Científica; lo cual permitió desarrollar la infraestructura de comunicaciones con fibra óptica, y establecer más enlaces satelitales hacia Cuernavaca, Mor., y San Pedro Mártir en Ensenada, Baja California Norte, a la par del primer enlace de microondas de alta velocidad entre la Torre II de Humanidades y la Dirección General de Servicios de Cómputo Académico, DGSCA, sobre la Ciudad de México.

Con esto último, se estableció en definitiva el final de la era del teleproceso, para dar paso a las redes de computadoras y sus enlaces a través de fibra óptica. En 1990, la UNAM fue la primera institución en Latinoamérica que se incorpora a la red mundial Internet, que enlaza a millones de máquinas y decenas de millones de usuarios en todo el mundo.

Dicha red es producto de un proyecto del Gobierno de los Estados Unidos que data de 1970, y en sus primeras etapas (como parte de un programa de investigación militar de ARPA), se logra demostrar la viabilidad de las comunicaciones entre computadoras, por medio de la conmutación de paquetes; lo cual creó la red ARPANET, que enlazó en sus primeros años varias decenas de sitios en una red nacional dedicada a la comunidad de investigación en computación.

El concepto de conmutación de paquetes se extendió en muy pocos años para incluir redes satelitales y redes basadas en radio. Su ininterrumpido desarrollo, que no tiene límite a la fecha, contempla como elemento fundamental el diseño de una arquitectura para comunicar redes que permitan la coexistencia de paquetes de comunicación de diferentes tipos, bajo el protocolo TCP/IP; éste último se mantiene como estándar en la actualidad, dada su funcionalidad y posibilidad de adaptación a los requerimientos que se van presentando. A finales de los 80s se da la apertura al uso comercial, en tanto se limitaba a proporcionar servicios a la comunidad académica.

Objetivos

- Dotar a la institución de una moderna infraestructura de telecomunicaciones y cómputo.
- Utilizar esta herramienta como un factor de transformación profundo en su modelo de enseñanza-aprendizaje.
- Proporcionar a su personal docente y de investigación todas las herramientas de la tecnología informática para el desarrollo de sus actividades.
- Integrar a sus alumnos, desde el bachillerato hasta el posgrado, a la cultura informática, entendida ésta como la integración del cómputo y las telecomunicaciones.
- Incorporar la enseñanza de la informática a los planes formales de estudio de todas las disciplinas, y actualizarla periódicamente.
- Difundir al público en general, los conocimientos y la tecnología que se genera en esta Casa de Estudios, resultado de investigaciones en las múltiples áreas del conocimiento.

Servicios

- Asesoría y soporte técnico al equipo de videoconferencia, a usuarios de la UNAM, y a externos.
- Claves personalizadas para extensiones de la UNAM larga distancia y celular.
- Facturación telefónica UNAM.
- Conexión a la red de la UNAM.
- Red de emergencia UNAM.
- Correo electrónico: servidor.unam.mx
- Cambio de password en servidor.unam.mx
- Llamadas a celular o larga distancia de extensiones de la UNAM.
- Asignación gratuita de acceso a Internet vía módem para académicos, funcionarios e instituciones en convenio.
- Ayuda para configurar el acceso a Internet (módem).
- Enlace satelital (datos).
- Enlace satelital (línea telefónica).
- Conectividad a Internet de una PC.
- Conectividad de una red local a Internet.
- Líneas directas de Telmex.
- Servicios digitales para líneas directas de Telmex.
- Cambio de claves personalizadas (códigos lada) para líneas directas de Telmex.
- Cambio de domicilio de líneas directas de Telmex.
- Llamadas a celular de líneas directas de Telmex.
- Claves personalizadas (códigos lada) para líneas directas de Telmex.
- Solicitud de teléfonos para casetas públicas móviles de Telmex y Telcel para eventos.
- Instalación de líneas directas y de extensiones de Telmex.
- Eventos que utilicen las líneas ISDN.
- Tratamiento de incidentes y quejas de seguridad.
- Servicio telefónico.

- Instalación de cableado telefónico interno y externo.
- Teléfono digital.
- Teléfono secretarial.
- Teléfonos unilínea.
- Servicios datos/voz/videoconferencia.
- Acceso a Internet vía módem.
- Asesoría en la instalación de enlaces para la conexión de aulas de videoconferencia.
- Asesoría técnica de cuentas de correo.
- Asignación, actualización y seguimiento de IPv6.
- Correo de los sitios que tienen Web Hosting.
- Actualización de datos.
- Avisos.

RedUNAM a la vanguardia tecnológica

Hoy en día, las tecnologías de la información y la comunicación constituyen una herramienta esencial para la producción, transmisión y difusión del conocimiento. La UNAM, al mantenerse actualizada respecto a la revolución tecnológica e incorporar constantemente nuevas tecnologías en su trabajo, conserva su liderazgo, que desde 1958 ha tenido en este ámbito, tanto en el país como en América Latina.

Por tal motivo, se trabaja para modernizar la administración de las telecomunicaciones de la Universidad, en un ambiente de convergencia tecnológica digital, con la finalidad de hacer compatible el avance y la innovación en términos de eficiencia y transparencia administrativa.

En el 2002, la DGSCA sustituyó la columna dorsal de la RedUNAM con un *backbone* capaz de soportar todos los servicios de datos, voz y video sobre IP y aplicaciones emergentes, garantizando a largo plazo, la disponibilidad de todos los servicios y el poder contar con una tecnología suficientemente flexible que permita el crecimiento cómo, dónde y cuándo se requiera.

Últimas innovaciones en materia de telecomunicaciones.

- Telecomunicación médica.
- Educación a distancia.
- Bibliotecas digitales.
- Telecomunicaciones submarinas.

Integración de Internet 2 a la RedUNAM.

Estadística

La RedUNAM atiende diariamente a 400 mil usuarios, de los cuales alrededor de 170 mil son parte de la Universidad, y los 230 mil restantes corresponden a conexiones de centros

de institutos de investigación, universidades públicas y privadas de todo el país, hospitales y dependencias de gobierno con las que la UNAM colabora.

El número de los servicios que la UNAM brinda en promedio mensual ascienden a más de 14 millones; en éstos se incluyen los hits al Web, los correos electrónicos enviados y recibidos, FTP y Telnet. Dichos servicios los otorga mediante los 31 nodos de cómputo y telecomunicaciones enlazados a través de mil 500 kilómetros tendidos de fibra óptica, por vía satelital y/o por microondas.

La Universidad cuenta con más de 38 mil computadoras, 30 mil puertos de red, 350 servidores Internet; 1,950 kilómetros de fibra óptica, dentro de los diversos campus universitarios, cien redes internas LAN, cien redes externas WAN; más de 70 mil cuentas de correo electrónico, laboratorios especializados para probar diversas tecnologías y su capacidad de interoperatividad; dos supercomputadoras capaces de realizar más de 20 mil millones de operaciones aritméticas por segundo, y más de dos mil estaciones de trabajo y almacenamiento central en disco superior a mil 500 millones de caracteres.

La UNAM posee la supercomputadora Kan Balam, la cual está compuesta por 1368 procesadores AMD y cuenta con 3016 GB de RAM y 160 TB de almacenamiento. Su capacidad de procesamiento es de 7.113 Teraflops, o bien, más de 7 billones de operaciones aritméticas por segundo. Está rankeada como la supercomputadora 28 a nivel mundial y la número 1 en América Latina.

RedUNAM permite más de 100 mil correos electrónicos recibidos por día; cerca de 500 mil visitas diarias a las diversas páginas de Internet y 200 mil usuarios de los servicios de información de la Universidad.

Hace más de 10 años se estableció el primer enlace de fibra óptica internacional, y hoy se tiene el primer enlace internacional comercial E3 a Internet, que complementa a múltiples enlaces E1 a Estados Unidos.

Recientemente entró en funcionamiento la primera conexión de Internet2 entre Ciudad Universitaria y el Centro de Supercómputo de San Diego, California, que permite enlaces de alta capacidad con las redes académicas de Estados Unidos, Canadá, Chile, Brasil y Argentina de manera directa.

La Universidad realiza comunicaciones académicas que la convierten en miembro distinguido y eje de comunidades virtuales en todas las especialidades.

Con sólo 2.5% de inversión adicional en 2008, en 13 dependencias universitarias, dentro de las que destacan todas las Facultades de Estudios Superiores, se amplió la capacidad de la conexión a RedUNAM e Internet, lo que significó un aumento de 30% en la capacidad con que se contaba en 2007.

Administración de RedUNAM

Para mantener un buen nivel de servicio en una red de dimensiones similares a las de RedUNAM, se requiere de un esquema de operación jerárquico; en el cual la Subdirección de Redes y Comunicaciones, de la DTD (Dirección de Telecomunicaciones Digitales), se ubica en el nivel más alto. Debajo de ésta, se encuentra el Departamento de Operación de Red, cuya tarea principal es la de administrar y operar la parte medular, o Backbone, de la red Universitaria y a su vez mantener operando la interconexión de todas las redes locales y enlaces dedicados -WAN- de las instituciones, organismos o empresas que se enlazan a RedUNAM.

El Departamento de Operación de la Red está integrado por:

- NOC-UNAM (Centro de Operación de RedUNAM).
- TAC-UNAM (Centro de Asistencia Técnica de RedUNAM)
- NIC-UNAM (Centro de Información de RedUNAM)

CENTRO DE OPERACIÓN DE RedUNAM (NOC-UNAM)

El NOC (Network Operation Center) o Centro de Operación de RedUNAM, es el encargado de mantener funcionando de manera eficiente la interconexión de las redes locales, los enlaces de área amplia y la "Columna Vertebral" o Backbone de la Red Universitaria.

El Centro de Operación proporciona apoyo a los administradores de redes locales para solucionar sus problemas de intercomunicación con otras redes; respondiendo así a la problemática que se presenta con el incremento del número de nodos en la red. Es importante señalar que el NOC también se encarga de estudiar el desempeño de la red y participar en las tareas de configuración, mantenimiento e implementación de las nuevas tecnologías en el backbone, manteniendo así a la Universidad a la vanguardia en las telecomunicaciones; tanto en servicio como en infraestructura.

Funciones

- **Monitoreo.**- El monitoreo es una de las partes fundamentales de la administración de las redes. Sus tareas comprenden, entre otras cosas, la extracción e interpretación de datos relacionados con el estado y el desempeño de los dispositivos conectados a la red. Mediante su correcta interpretación, y llevando un registro histórico de los acontecimientos que se van dando a lo largo del tiempo, el administrador de red podrá determinar de manera más rápida el comportamiento de ciertos equipos e incluso, adelantarse y predecir el deterioro del nivel de servicio en alguna parte de la red. Para poder realizar esta tarea, el NOC cuenta con sistemas de monitoreo que trabajan las 24 hrs del día bajo el ambiente cliente-servidor, permitiendo manejar de manera distribuida el tráfico generado, de manera natural, por los sistemas de monitoreo tradicionales.
- **Tarificación.**- Se refiere a contabilizar el tráfico generado por todas las redes locales que se conectan a RedUNAM. Los sistemas de monitoreo recolectan esta

información diariamente para ser almacenada en una base de datos que será accesada por programas desarrollados en el NOC, con la finalidad de generar de manera condensada la contabilidad de tráfico de cada red local.

- **Generación de estadísticas.**- Con el fin de informar a los administradores de red, el desempeño de sus enlaces y la capacidad con la que se están ocupando. Para que de esta manera, cada administrador pueda contar con un historial del desempeño y ocupación de su enlace; lo cual contribuye a la correcta planeación de sus crecimientos. Se utilizan herramientas que permiten presentar los registros en tiempo real de forma gráfica. Se presentan únicamente algunas de las variables más importantes a estudiar en el desempeño de una red o de un enlace dedicado.
- **Administración de equipos de ruteo.**- Estos dispositivos son los encargados de dirigir la información de un lado a otro por la ruta más óptima. Debido a que los ruteadores conectados a RedUNAM son de diferentes marcas, cada uno puede manejar uno o más protocolos de ruteo, entre los cuales podemos mencionar a RIP, IGRP, EIGRP, OSPF, BGP, etc.

Una vez conectados a la infraestructura de la red, el administrador deberá configurarlos de tal manera que puedan intercambiar entre ellos "información de ruteo", para que en conjunto construyan una serie de alternativas de flujo de información hacia diferentes destinos; que generalmente son las estaciones finales que están corriendo las aplicaciones del usuario.

Por otro lado, la administración de estos dispositivos contempla, también, el manejo de sistemas de seguridad como "firewalls" y listas de acceso. Para esto el NOC de la UNAM, como muchos otros NOC's en el mundo, cuenta con sistemas que guardan un registro de los usuarios que establecieron comunicación con nuestros equipos de ruteo, ya que naturalmente este tipo de accesos deberán ser restringidos. Así pues, solo el personal del NOC puede tener acceso a los mismos para hacer cambios en las configuraciones, asegurando así su buen funcionamiento.

- **Seguimiento de reportes (Troubleshooting).**- El NOC de RedUNAM realiza como una actividad cotidiana, la atención de reportes de fallas. Una falla en la red, significa que algún dispositivo dentro de la misma no se desempeña como se espera y que por lo tanto se le debe poner atención hasta que se normalice su funcionamiento. El NOC se encarga de minimizar el número de fallas y reducir el tiempo de resolución de las mismas.

Las actividades básicas a desarrollar durante el seguimiento de reportes son:

- Detección de la falla
- Diagnóstico de la falla
- Determinación de la falla
- Resolución de la falla

Para la detección, se cuenta con sistemas que despliegan en monitores, de manera gráfica o textual, los problemas existentes en la red. O bien, para cuando el staff del NOC no se encuentra atendiendo a sus consolas, se utiliza un sistema que "redirecciona" estos mensajes de alarma a los correos electrónicos o inclusive a los localizadores de los "troubleshooters".

Con lo anterior, el NOC de RedUNAM está las 24 Hrs del día, los 365 días del año, al tanto del estado de los dispositivos más importantes que proporcionan el servicio de la red de datos en la UNAM.

El diagnóstico de la falla se realiza de manera un poco más compleja. Aquí intervienen una serie de herramientas de prueba -tanto de software como de hardware- que proporcionan la información necesaria para determinar qué es lo que está provocándola. Para tener la idea precisa de lo que se trata -determinación de la falla- cuenta mucho la experiencia del staff del NOC y de la ayuda de los sistemas de manejo de reportes, que llevan la historia de los casos resueltos con anterioridad. Con esto, se garantiza la rápida solución de los problemas que se presentan comúnmente en las redes de datos.

- **Documentación tecnológica.**- Representa información técnica especializada, orientada a las personas relacionadas con el manejo de redes de datos y contribuye a la formación profesional de los integrantes del NOC.

CENTRO DE ASISTENCIA TÉCNICA DE RedUNAM (TAC-UNAM)

El Centro de Asistencia Técnica de la UNAM (TAC-UNAM) tiene como objetivo primordial garantizar el buen funcionamiento de la Red de datos de la UNAM: RedUNAM y optimizar la operación de ésta, llevando a cabo la administración de los equipos activos de comunicación como son: switches de backbone, switches periféricos y switches de acceso, proporcionando así el servicio a todas las redes LAN de las dependencias distribuidas en los campus universitarios. De igual manera es el encargado de proporcionar a la comunidad universitaria y a usuarios en general el Servicio de Acceso Remoto también conocido como Servicio Dial-Up, permitiendo realizar una conexión confiable y segura a RedUNAM desde cualquier ubicación geográfica.

Con la finalidad de sustentar la operación y crecimiento de la red de datos de RedUNAM, el TAC-UNAM proporciona los siguientes servicios para las Dependencias Internas de la universidad conectadas en línea directa.

Servicios

- Instalaciones Programadas.
- Asesoría técnica a los Administradores de Red.
- Instalación y/o configuración de tarjetas de red.
- Configuración del protocolo TCP/IP para Windows 95/NT y equipos UNIX.
- Asignación y/o instalación de software de RedUNAM.
- Detección y corrección de fallas en las redes de la UNAM.
- Análisis de redes.
- Visitas a RedUNAM.
- Herramientas para prueba.

CENTRO DE INFORMACIÓN DE RedUNAM (NIC-UNAM)

Un NIC (Network Information Center), de acuerdo a los esquemas de Internet, es un organismo encargado de la administración de rangos o bloques de direcciones IP y dominios.

RedUNAM está compuesta por un conjunto de redes locales que tienen una administración propia, pero al estar conectadas a toda la red sus encargados y usuarios deben acatar las disposiciones establecidas por la Dirección de Telecomunicaciones con representación por parte del NIC-UNAM, lo que representa un esquema de administración jerárquica.

El Centro de Información de RedUNAM (NIC-UNAM) se encarga de proporcionar servicios de: Asignación de Direcciones IP, Asignación de Dominios y Servicio de Nombres a las dependencias e instituciones conectadas a RedUNAM. Como actividad primordial, el NIC-UNAM se encarga del mantenimiento de una Base de Datos donde se concentra información concerniente a la administración de Red e Instituciones.

Las funciones principales son:

- Administración de los rangos y direcciones IP que pertenecen a la UNAM. Esto incluye su correcta distribución y asignación a las dependencias que así lo requieran.
- Administración de los servidores de nombres encargados de la resolución de dominios asociados a direcciones IP o hosts de la UNAM. Entre ellos tenemos principalmente:
 - unam.mx (y todos sus subdominios)
 - 248.132.in-addr.arpa
- Establecer las políticas para cada uno de los servicios que ofrece RedUNAM a todos sus usuarios.
- Ser el organismo de representación oficial de la UNAM ante los organismos Regionales, Nacionales e Internacionales que tienen relación directa con Internet.

Servicios

Los servicios que ofrece NIC-UNAM están orientados básicamente a las necesidades generales de cada una de las redes locales como son:

- **Servicio de Nombres.**- Actualmente todos los hosts cuentan con una dirección IP y un nombre, el cual está asociado al dominio UNAM.MX. Esto les permite ser identificados dentro de Internet y poder tener comunicación con otros hosts, sobre todo en aplicaciones como el correo electrónico. Por ello es conveniente que cada uno de ellos esté dado de alta en las Bases de Datos del Servicio de Nombres; esto se hace mediante una solicitud que envía el administrador de la red local al NIC-UNAM.
- **Asignación de Direcciones IP.**- El esquema de direccionamiento que se utiliza en RedUNAM está basado en direcciones IP. El NIC-UNAM asigna a cada una de las dependencias que se conectan a la red un segmento o un rango de direcciones IP; el

administrador de red local es el responsable de hacer esta solicitud al NIC-UNAM. Una vez hecha la asignación, esta persona es la encargada de asignar una dirección IP a cada uno de los hosts que conforman la red de la dependencia. De la misma forma se asignan clases de direcciones IP a instituciones independientes a la UNAM que se conectan a la red universitaria, y que así lo requieren.

- **Asignación y Solicitud de Dominios.**- En el caso de Dependencias de la UNAM, el NIC le asigna a cada una de ellas un subdominio bajo UNAM.MX (p.ej.: tic.unam.mx) previa solicitud del administrador local. Para las instituciones externas, el NIC de la UNAM hace la solicitud de dominios ante NIC-México (<http://www.nic.mx>) para aquellas que así lo requieran.
- **Servicio de Servidor Secundario.**- En muchas ocasiones, instituciones ajenas a la Universidad requieren tener un servidor secundario para sus dominios; el NIC-UNAM ofrece este servicio a aquellas que están directamente conectadas a RedUNAM.
- **Conexión con RedUNAM.**- El NIC-UNAM en conjunto con el NOC-UNAM, ofrecen un servicio de información y soporte técnico a aquellas instituciones que deseen hacer su conexión con RedUNAM para integrarse a Internet.

Asimismo, desempeña las siguientes actividades:

- **Capacitación.**- El NIC-UNAM coordina gran parte de las actividades dirigidas a la capacitación de los administradores de red de la UNAM, por medio de seminarios, cursos y talleres siempre buscando una formación de recursos humanos que permita brindar un soporte de calidad a sus usuarios.
- **Elaboración e implementación de políticas.**- Este es uno de los pilares para la administración de RedUNAM. El NIC-UNAM elabora políticas concernientes a cada uno de los servicios que brinda y de acuerdo a las necesidades de administración. Este trabajo se lleva a cabo en conjunto con los diversos grupos de trabajo que intervienen en la operación de la red universitaria. Por medio de los diferentes servicios de información da a conocer a los administradores de red y a la comunidad universitaria los lineamientos que se deben seguir para hacer uso de servicios, petición de direcciones IP, solicitud de dominios, etc. Los administradores de red deben de estar de acuerdo con estas políticas y es su responsabilidad hacerlas del conocimiento de sus usuarios.
- **Eventos.**- El NIC-UNAM coordina la realización de eventos relacionados a RedUNAM que sirven para dar a conocer las actividades que se llevan a cabo en el ámbito de la red universitaria. Estos eventos se realizan en común acuerdo con los administradores de redes locales de la UNAM y otras instituciones, o por iniciativa del NIC-UNAM.
- **Presencia en organismos nacionales e internacionales.**- Dada la investigación y desarrollo que se hace en la UNAM, el NIC-UNAM estará presente en los organismos nacionales e internacionales, sobre todo aquellos relativos a Internet, con el fin de aportar nuevas ideas y proyectos.

Dentro de Internet encontramos organismos como: ISOC (Internet Society), IETF (Internet Engineering Task Force), CERT (Computer Emergency Response Team); cada uno de ellos tiene en común la investigación y desarrollo en diferentes ámbitos y buscan soluciones a diversos problemas.

BASE DE DATOS DE NIC-UNAM

En la actualidad, NIC-UNAM posee una base de datos elaborada con el manejador de bases de datos Sybase, la cual tiene una interfaz gráfica llamada Action Request System 2.1 (elaborada por Remedy Corporation). Es una aplicación de escritorio instalada en diferentes clientes, donde se efectúan todas las consultas correspondientes a la red de datos de la UNAM (RedUNAM), la cual está formada por las siguientes redes:

- 132.247.0.0/16
- 132.248.0.0/16

Al día de hoy el segmento 248 ha sido utilizado al 90% de su capacidad y se han empezado a asignar algunas redes del segmento 247.

Se ha realizado un análisis exhaustivo de dicha base de datos tomando en cuenta diferentes parámetros tales como: desempeño, rapidez, integridad, seguridad, capacidad de almacenamiento, entre otros.

A continuación se mencionan varios aspectos que se han encontrado a partir del análisis realizado:

- Para poder tener acceso a la base de datos, se necesita instalar previamente la aplicación en cada computadora donde será utilizada. La máquina debe cumplir con ciertos requerimientos de hardware y software así como seguir un proceso de instalación diferente dependiendo de la arquitectura y/o sistema operativo de dicha máquina, aunque también se han presentado algunos casos donde ha sido imposible finalizar la instalación con éxito. Cabe mencionar que entre los requerimientos más importantes a considerar se encuentran: la memoria RAM, la capacidad en disco duro y el procesador ya que la ejecución de la aplicación consume muchos recursos de la computadora. Todos estos factores que intervienen en la instalación hacen que la consulta de información se vuelva un trabajo arduo, complicado y muy particular que tiene como consecuencia la pérdida de tiempo, variable valiosa para el usuario.
- El software que se utiliza es propietario (Sybase) por lo que se tiene determinado número de licencias, es decir, un número finito de máquinas cliente donde puede consultarse la información de la base de datos, lo cual limita el acceso a un número finito de personas en máquinas específicas ubicadas geográficamente en el mismo lugar; además, existen muchos riesgos a los cuales se encuentran expuestas dichas computadoras, ya sea por problemas técnicos, físicos, humanos, geográficos, etc., que en algún momento dado pueden impedir el acceso a la base de datos.

- El servidor se encuentra instalado en una máquina con las siguientes características:
 - Sistema operativo: SunOS 5.7 (Solaris 7)
 - Plataforma: UltraSPARC-III
 - Velocidad: 360MHz
 - Capacidad de memoria: 128 Megabytes

Actualmente, dicho sistema operativo ya no se distribuye pero aún está soportado. Debido a que no se ha actualizado la versión del sistema operativo, la seguridad no es muy alta, además podemos encontrar deficiencias en cuanto al desempeño debido a las características tanto de software como de hardware de la máquina. De software en: actualizaciones, parches, interfaz, programas y compatibilidad de éstos; de hardware en: velocidad, capacidad en disco duro y memoria RAM. Por lo que es recomendable migrar la base de datos a una nueva máquina y actualizar la versión del sistema operativo, o bien, utilizar otro, a lo que se presentan diversos inconvenientes como compatibilidad en archivos, comandos distintos, entre otros.

- RedUNAM está en constante cambio y hoy en día tiene una nueva distribución, lo cual hace que la base de datos actual quede limitada ya que se han agregado nuevos campos que implican modificaciones y nuevas tablas. Pero después de haber realizado el análisis correspondiente al diseño de la base de datos y obtener cada una de las tablas, se ha llegado a la conclusión de que el hecho de modificarlas o agregar nuevas tablas implica un proceso muy complicado y en algunos casos imposible ya que durante su etapa de diseño no se tomaron en cuenta las posibilidades de expansión a futuro ni las mejoras que se le podían hacer a dicha base, es decir, la base no es escalable.
- El aumento de la población universitaria ha generado la construcción de nuevos edificios, aulas y centros de cómputo. Asimismo, la ampliación y extensión de ciertas dependencias universitarias. Sin dejar de mencionar que nuevas dependencias externas se han enlazado a RedUNAM. Todo esto produce el aumento de las redes y que la demanda de éstas siga incrementándose día con día, los cuales son factores determinantes que han modificado los requerimientos de la base de datos. Ahora se necesita un sistema de bases de datos que cuente con más capacidad y se adapte a las necesidades que se vayan presentando para que pueda ofrecer un mejor desempeño.
- Además de la administración de la red de datos también se incluirá la red de voz, por lo que se modifica en su totalidad el esquema que se tenía planteado en un principio. La red de voz comparte información con la red de datos pero es preciso aclarar que difiere en algunas secciones, además de tener casos particulares con datos específicos de ésta.
- Los datos que se manejan en esta base de datos son privados y muy importantes, son datos confidenciales a los cuales sólo determinados usuarios deben tener acceso, y sólo una persona debería ser el administrador, ya que de esto dependerá tanto la

eficacia como la seguridad. Actualmente son varios los usuarios que administran dicha base de datos o lo han hecho cuando se ha presentado algún problema, situación que la hace vulnerable. Es necesario plantear políticas de seguridad y sobre todo definir y establecer los privilegios para cada uno de los usuarios del sistema.

Por lo anteriormente mencionado, se puede concluir que se ha encontrado que la base de datos actual presenta las siguientes deficiencias y errores:

- No portable.
- No escalable.
- Dificultad en la instalación de la aplicación.
- Mal diseño en su estructura interna.
- Interfaz poco amigable.
- Muy particular.
- Utiliza muchos recursos de la computadora en su ejecución.
- No cubre todos los requerimientos (no abarca la red de voz).
- Poca seguridad.
- Costo de la licencia de Sybase por usuario.
- Aplicación de escritorio instalada en determinadas máquinas.
- El servidor se encuentra instalado en máquina antigua.
- Información incompleta debido a que el esquema de RedUNAM ha cambiado.
- Poca capacidad de almacenamiento.

Estos son los motivos por los cuales se ha decidido crear un sistema de bases de datos que elimine dichos errores y brinde nuevos servicios que den respuesta a los requerimientos actuales y satisfaga por completo las necesidades de los usuarios, con un mejor desempeño y que su diseño permita ser modificado (escalable) si se presenta el caso ya que RedUNAM está en constante cambio y por lo tanto el sistema debe ser lo suficientemente flexible y adaptable. Asimismo, cabe destacar el hecho de que crear un sistema de bases de datos es lo más óptimo y recomendable debido a que modificar y migrar la base de datos que se tiene actualmente es una tarea sumamente complicada y además no se podrían cumplir los objetivos en su totalidad ni se brindaría la calidad necesaria por lo que el producto final estaría incompleto.

La nueva base de datos será de gran importancia ya que es pieza fundamental para tener un registro, control adecuado y una buena administración de las redes que integran RedUNAM. Esta nueva base de datos reemplazará la que ya se tiene y resolverá todos los problemas antes mencionados y brindará nuevos servicios, no sólo como una herramienta de acceso a la información sino que será todo un sistema de bases de datos.

CAPÍTULO 1

CONCEPTOS BÁSICOS

1.1 BASES DE DATOS

Una base de datos es un conjunto de datos relacionados, o bien, datos con características similares. Dicha base de datos tiene las siguientes propiedades implícitas:

- Una base de datos representa algunos aspectos del mundo real, en ocasiones denominado minimundo o Universo del Discurso (UdD). Los cambios en el minimundo se reflejan en la base de datos.
- Una base de datos es una colección coherente de datos con significados inherentes, por lo que un conjunto aleatorio de datos no puede considerarse como una base de datos.
- Una base de datos se diseña, construye y puebla con datos para un propósito específico. Está destinada a un grupo de usuarios concreto y tiene algunas aplicaciones preconcebidas en las cuales están interesados dichos usuarios.

En otras palabras, una base de datos tiene alguna fuente de la cual provienen los datos, algún grado de interacción con los sucesos del mundo real, y una audiencia que está activamente interesada en el contenido de la base de datos.

En esencia, una base de datos no es más que una colección de información que existe a lo largo de un período de tiempo. Más claramente, el término base de datos se refiere a una colección de datos sin redundancia gestionada por un Sistema Gestor de Bases de Datos, S.G.B.D.

Sistema de Gestión de Bases de Datos

Un sistema de gestión de bases de datos (*SGDB, en inglés DataBase Management System o DBMS*) es una colección de programas que permiten a los usuarios crear, mantener y tener acceso a una base de datos. El SGBD es por tanto un sistema software de propósito general que facilita los procesos de definición, construcción y manipulación de bases de datos para distintas aplicaciones.

- La definición de una base de datos consiste en especificar los tipos de datos, las estructuras y restricciones para los datos que se van a almacenar en dicha base.
- La construcción de la base de datos es el proceso de almacenar los datos concretos sobre algún medio de almacenamiento controlado por el SGBD.
- La manipulación de la base de datos incluye funciones tales como consultar la base de datos para recuperar unos datos específicos, actualizar la base de datos para reflejar los cambios ocurridos en el minimundo, y generar informes a partir de los datos.

No. de Servicio	Descripción
1	Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un SGBD y por supuesto, el SGBD debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).
2	Un SGBD debe proporcionar un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios. Este catálogo es lo que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos).
3	Un SGBD debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos.
4	Un SGBD debe proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es el permitir que varios usuarios tengan acceso concurrente a los datos que comparten.
5	Un SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe. Cuando el sistema falla en medio de una transacción, la base de datos se debe devolver a un estado consistente.
6	Un SGBD debe proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.
7	Un SGBD debe ser capaz de integrarse con algún software de comunicación.
8	Un SGBD debe proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realizan sobre estos datos, sigan ciertas reglas. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados. Se puede considerar como otro modo de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones. La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar.

Tabla 1.1 Funciones de los SGBD (Codd, 1970).

Además, se espera que todo SGBD proporcione los siguientes servicios:

1. Un SGBD debe permitir que se mantenga la independencia entre los programas y la estructura de la base de datos. La independencia de datos se alcanza mediante las vistas o subesquemas.
2. Un SGBD debe proporcionar una serie de herramientas a nivel externo (producidas por el administrador de la BD) e interno (proporcionadas por el distribuidor del SGBD) que permitan administrar la base de datos de modo efectivo

Sistema de Bases de Datos

Un Sistema de Bases de Datos consiste en una colección de datos interrelacionados y un conjunto de programas de aplicación para acceder y/o modificar dichos datos. Por lo tanto, se denominará Sistema de Bases de Datos al conjunto formado por la base de datos más el SGBD. Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información.

Ventajas	Desventajas
<p>Por la integración de datos:</p> <ul style="list-style-type: none"> • Control sobre la redundancia de datos. • Consistencia de datos. • Más información sobre la misma cantidad de datos. • Compartición de datos. • Mantenimiento de estándares. • Aislamiento de datos. <p>Por la existencia del SGBD:</p> <ul style="list-style-type: none"> • Mejora en la integridad de datos. • Mejora en la seguridad. • Mejora en la accesibilidad a los datos. • Mejora en la productividad. • Mejora en el mantenimiento gracias a la independencia de datos. • Aumento de la concurrencia. • Mejora en los servicios de copias de seguridad y de recuperación ante fallos. 	<ul style="list-style-type: none"> • Complejidad. • Personal capacitado. • Tamaño. • Coste económico del SGBD. • Coste del equipamiento adicional. • Coste de la conversión. • Prestaciones. • Vulnerable a los fallos.

Tabla 1.2 Ventajas y desventajas de un sistema de bases de datos.

Clasificación de los sistemas de bases de datos

El criterio principal que se utiliza para clasificar los SGBD es el modelo lógico en que se basan. Los modelos lógicos empleados con mayor frecuencia en los SGBD comerciales actuales son el *relacional*, el de *red* y el *jerárquico*. Algunos SGBD más modernos se basan en modelos orientados a objetos.

Un segundo criterio para clasificar los SGBD es el número de usuarios a los que da servicio el sistema. Los sistemas *monousuario* sólo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales. Los sistemas *multiusuario*, entre los que se encuentran la mayor parte de los SGBD, atienden a varios usuarios al mismo tiempo.

Un tercer criterio es el número de sitios en los que está distribuida la base de datos. Casi todos los SGBD son *centralizados*, es decir, sus datos se almacenan en un solo computador. Los SGBD centralizados pueden atender a varios usuarios, pero el SGBD y la base de datos en sí residen por completo en una sola máquina. En los SGBD *distribuidos* la base de datos real y el propio software del SGBD pueden estar distribuidos en varios sitios conectados por una red. Los SGBD *distribuidos homogéneos* utilizan el mismo SGBD en múltiples sitios. Una tendencia reciente consiste en crear software para tener acceso a varias bases de datos autónomas preexistentes almacenadas en SGBD *distribuidos heterogéneos*. Esto da lugar a los SGBD *federados* o *sistemas multibase de datos* en los que los SGBD participantes tienen cierto grado de autonomía local. Muchos SGBD distribuidos emplean una arquitectura cliente-servidor.

Un cuarto criterio es el coste del SGBD.

Por último, los SGBD pueden ser de *propósito general* o de *propósito específico*. Cuando el rendimiento es fundamental, se puede diseñar y construir un SGBD de propósito especial para una aplicación específica, y este sistema no sirve para otras aplicaciones.

Arquitecturas de los sistemas de bases de datos

La arquitectura de un sistema de bases de datos está influenciada en gran medida por el sistema informático subyacente en el que se ejecuta, en particular por aspectos de la arquitectura de la computadora como la conexión en red, el paralelismo y la distribución como se describe a continuación:

- La conexión en red de varias computadoras permite que algunas tareas se ejecuten en un sistema servidor y que otras se ejecuten en los sistemas clientes. Esta división de trabajo ha conducido al desarrollo de *sistemas de bases de datos cliente-servidor*.
- El procesamiento paralelo dentro de una computadora permite acelerar las actividades del sistema de base de datos, proporcionando a las transacciones unas respuestas más rápidas así como la capacidad de ejecutar más transacciones por segundo. Las consultas pueden procesarse de manera que se explote el paralelismo ofrecido por el sistema informático subyacente. La necesidad del procesamiento paralelo de consultas ha conducido al desarrollo de los *sistemas de bases de datos paralelos*.
- La distribución de datos a través de las distintas sedes o departamentos de una organización permite que estos datos residan donde han sido generados o donde son más necesarios, pero continuar siendo accesibles desde otros lugares o departamentos diferentes. Se han desarrollado los *sistemas distribuidos de bases de datos* para manejar datos distribuidos geográfica o administrativamente a lo largo de múltiples sistemas de bases de datos.

1.1.1 CLIENTE – SERVIDOR

La mayoría de usuarios de un sistema de bases de datos no están situados actualmente junto al sistema de bases de datos, sino que se conectan a él a través de una red. Se puede diferenciar entonces entre las máquinas cliente, en donde trabajan los usuarios remotos de la base de datos, y las máquinas servidor, en las que se ejecuta el sistema de bases de datos. En la siguiente figura se representa la estructura general de un sistema cliente-servidor.

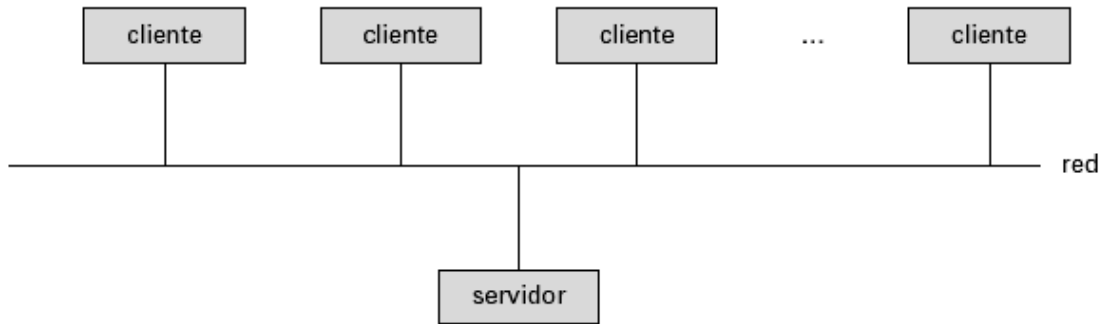


Figura 1.1 Estructura general de un sistema cliente – servidor.

Como se muestra en la figura 1.2, la funcionalidad de una base de datos se puede dividir a grandes rasgos en dos partes: la parte visible al usuario y el sistema subyacente. El sistema subyacente gestiona el acceso a las estructuras, la evaluación y optimización de consultas, el control de concurrencia y la recuperación. La parte visible al usuario de un sistema de base de datos está formado por herramientas como formularios, diseñadores de informes y facilidades gráficas de interfaz de usuario. La interfaz entre la parte visible al usuario y el sistema subyacente puede ser SQL o una aplicación.

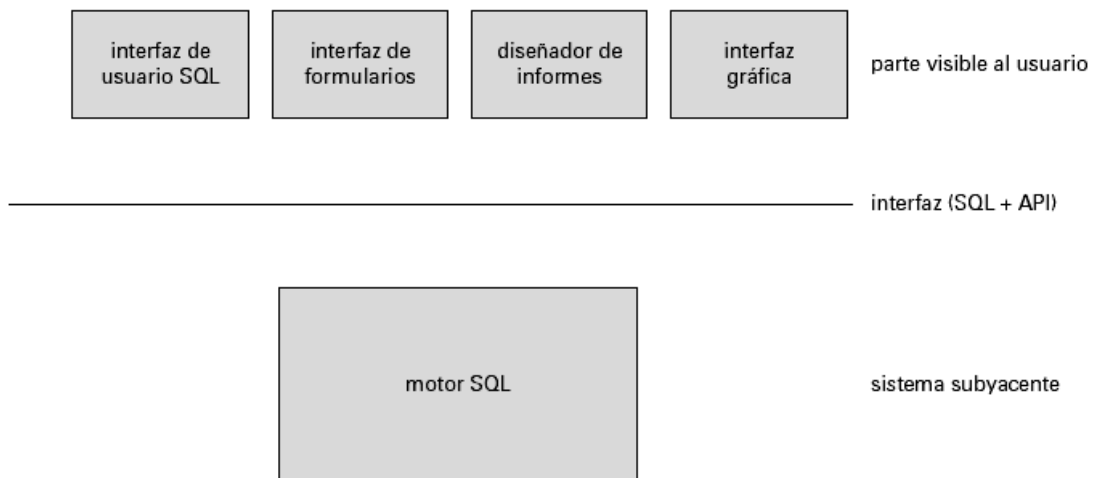


Figura 1.2 Funcionalidades de la parte visible al usuario y del sistema subyacente.

Las aplicaciones de bases de datos se dividen usualmente en dos o tres partes. En una arquitectura de dos capas, la aplicación se divide en un componente que reside en la

máquina cliente, que llama a la funcionalidad del sistema de bases de datos en la máquina servidor mediante instrucciones del lenguaje de consultas. Los estándares de interfaces de programas de aplicación como ODBC y JDBC se usan para la interacción entre el cliente y el servidor. En cambio, en una arquitectura de tres capas, la máquina cliente actúa simplemente como frontal y no contiene ninguna llamada directa a la base de datos. En su lugar, el cliente se comunica con un servidor de aplicaciones, usualmente mediante una interfaz de formularios. El servidor de aplicaciones, a su vez, se comunica con el sistema de bases de datos para acceder a los datos.

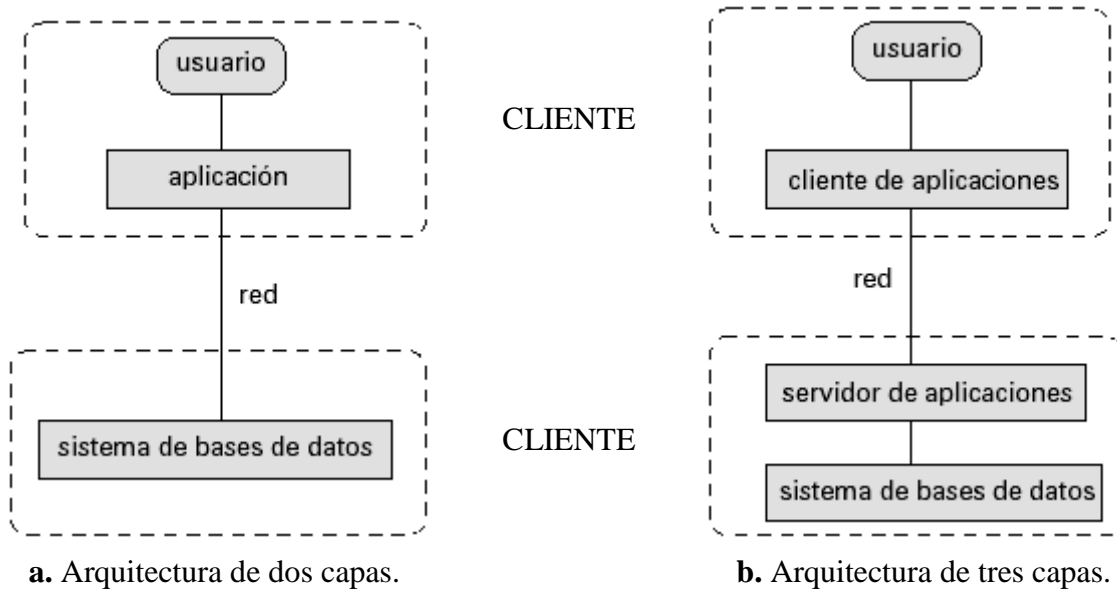


Figura 1.3 Arquitectura de dos y tres capas.

1.1.2 NIVELES DE ABSTRACCIÓN DE DATOS

Nivel físico o interno.- El nivel más bajo de abstracción describe cómo se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.

Nivel lógico o conceptual.- El siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples. La implementación de estructuras simples en el nivel lógico involucra estructuras complejas del nivel físico.

Nivel de vistas o externo.- El nivel más alto de abstracción describe sólo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. Muchos usuarios necesitan acceder sólo a una parte de la base de datos. Para que su interacción con el sistema se simplifique, se define la abstracción del nivel de vistas. El sistema puede proporcionar muchas vistas para la misma base de datos.

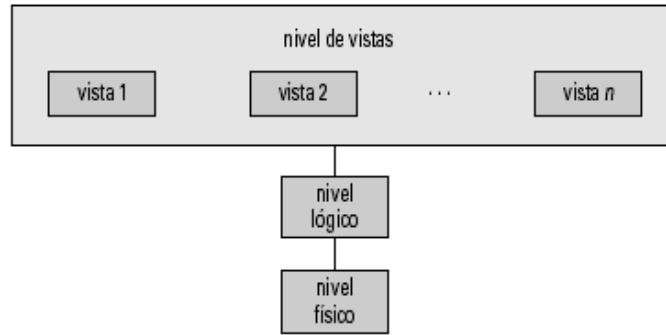


Figura 1.4 Los tres niveles de abstracción de los datos.

Ejemplares y esquemas

Se denomina ejemplar de la base de datos a la colección de información almacenada en la base de datos en un momento particular. Al diseño completo de la base de datos se le llama esquema de la base de datos.

Los sistemas de bases de datos tienen varios esquemas divididos de acuerdo a los niveles de abstracción. El esquema físico describe el diseño físico en el nivel físico, mientras que el esquema lógico describe el diseño de la base de datos en el nivel lógico. Una base de datos puede tener también varios esquemas en el nivel de vistas, a menudo denominados subesquemas, que describen diferentes vistas de la base de datos.

Independencia de datos

La independencia de datos se define como la capacidad para modificar el esquema en un nivel del sistema de bases de datos sin tener que modificar el esquema del nivel inmediato superior. Existen dos tipos de independencia de datos:

1. **Independencia lógica de los datos.**- es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede cambiar la estructura de datos almacenados, sin afectarlos.
2. **Independencia física de los datos.**- es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Las aplicaciones permanecen inalteradas sin importar los cambios efectuados en el almacenamiento o en los métodos de acceso.

1.1.3 LENGUAJES DE BASES DE DATOS

Lenguaje de Definición de Datos o DDL

Un esquema de base de datos se especifica mediante un conjunto de definiciones expresadas mediante un lenguaje especial llamado lenguaje de definición de datos (DDL). Ejemplos de SQL: create, alter, drop.

Diccionario de Datos o DD

Un diccionario de datos o directorio de datos es un conjunto especial de tablas que contiene metadatos, es decir, datos acerca de los datos. Un sistema de base de datos consulta el diccionario de datos antes de leer o modificar los datos reales. Normalmente, un diccionario de datos almacena:

- Nombre, tipo y tamaño de los datos.
- Nombre de las relaciones entre los datos.
- Restricciones de integridad sobre los datos.
- Nombre de los usuarios autorizados a acceder a la base de datos.
- Esquemas externo, conceptual e interno, y correspondencia entre los esquemas.
- Estadísticas de utilización, tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.

Lenguaje de Manipulación de Datos o DML

Lenguaje de manipulación de datos (LMD).- es un lenguaje que permite a los usuarios acceder o manipular los datos organizados mediante el modelo de datos apropiado. Hay dos tipos básicamente:

- *Procedimentales.*- Requieren que el usuario especifique qué datos se necesitan y cómo obtener esos datos.
- *Declarativos.*- (también conocidos como *no procedimentales*). Requieren que el usuario especifique qué datos se necesitan sin especificar cómo obtener esos datos.

Una consulta es una instrucción de solicitud para recuperar información. La parte de un LMD que implica recuperación de información se llama lenguaje de consultas, o bien, lenguaje de manipulación de datos.

SQL es el lenguaje de consultas más ampliamente usado aunque cabe resaltar que existen otros lenguajes de consultas. Ejemplos de SQL: insert, update, select, delete.

Lenguaje de Control de Datos o DCL

Permite la asignación de los diferentes permisos sobre los objetos existentes en una base de datos. Permite la definición de lo que los usuarios pueden hacer dentro de la base de datos. Ejemplos de SQL: grant, revoke.

1.1.4 SQL

El lenguaje de consulta estructurado (SQL) es un lenguaje declarativo y normalizado de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar de manera sencilla la información de interés de una base de datos, así como también hacer cambios u operaciones sobre la

misma, ya sea en los datos, o bien, en la estructura. Es un lenguaje de cuarta generación (4GL).

Los orígenes del SQL están ligados a los de las bases de datos relacionales. En 1970 E. F. Codd propone el modelo relacional y asociado a este un sublenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas, los laboratorios de IBM definen el lenguaje SEQUEL (Structured English QUery Language) que más tarde sería ampliamente implementado por el SGBD experimental System R, desarrollado en 1977 también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en 1979 en un programa comercial.

El SEQUEL terminaría siendo el predecesor de SQL, siendo éste una versión evolucionada del primero. El SQL pasa a ser el lenguaje por excelencia de los diversos SGBD relacionales surgidos en los años siguientes y es por fin estandarizado en 1986 por el ANSI, dando lugar a la primera versión estándar de este lenguaje, el SQL-86 o SQL1. Al año siguiente este estándar es también adoptado por la ISO. Sin embargo este primer estándar no cubre todas las necesidades de los desarrolladores e incluye funcionalidades de definición de almacenamiento que se consideraron suprimir. Así que en 1992 se lanza un nuevo estándar ampliado y revisado del SQL llamado SQL-92 o SQL2.

Componentes del SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Comandos

Existen dos tipos de comandos SQL:

- DDL que permiten crear y definir nuevas bases de datos, campos e índices.
- DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comando	Descripción
CREATE	Crea nuevas tablas, campos e índices
DROP	Elimina tablas e índices
ALTER	Modifica tablas agregando campos o cambiando la definición de éstos

Tabla 1.3 Comandos DDL.

Comando	Descripción
SELECT	Para consultar registros de la base de datos que satisfagan un criterio determinado.
INSERT	Para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Para modificar los valores de los campos y registros especificados.
DELETE	Para eliminar registros de una tabla de una base de datos.

Tabla 1.4 Comandos DML.

Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Cláusula	Descripción
FROM	Para especificar la tabla de la cual se van a seleccionar los registros.
WHERE	Para especificar las condiciones que deben reunir los registros que se van a seleccionar.
GROUP BY	Para separar los registros seleccionados en grupos específicos.
HAVING	Para expresar la condición que debe satisfacer cada grupo.
ORDER BY	Para ordenar los registros seleccionados de acuerdo con un orden específico.

Tabla 1.5 Cláusulas.

Operadores Lógicos

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

Tabla 1.6 Operadores Lógicos.

Operadores de Comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor o igual que
>=	Mayor o igual que
=	Igual que
BETWEEN	Para especificar un intervalo de valores
LIKE	En la comparación de un modelo
IN	Para especificar registros en una base de datos

Tabla 1.7 Operadores de comparación.

MySQL y PostgreSQL son los principales SGBD de uso libre y con código fuente. Es por esta razón que a continuación se presenta cada uno de ellos:

❖ MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

MySQL AB (desde enero de 2008 una subsidiaria de Sun Microsystems) desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

MySQL está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

Sus principales objetivos han sido la velocidad y la robustez, por lo que se puede decir que es un SGBD sencillo y rápido que se adapta perfectamente a entornos en donde el volumen de datos es del orden de megabytes.

Una de las cosas que más se agradecen de MySQL es que las bases de datos las almacena creando un directorio por cada una de ellas, y dentro de dicho directorio, crea tres ficheros por tabla donde se almacenan los registros de la tabla y la definición de la tabla. El copiar una base de datos es tan fácil como copiar un directorio por lo que hacer copias de seguridad de las mismas es algo de lo más sencillo.

Lenguajes de programación

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP.

Aplicaciones

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Plataformas

MySQL funciona sobre múltiples plataformas, incluyendo:

- AIX
- BSD
- FreeBSD
- HP-UX
- GNU/Linux
- Mac OS X
- NetBSD
- Novell Netware
- OpenBSD
- OS/2 Warp
- QNX
- SGI IRIX
- Solaris
- SunOS
- SCO OpenServer
- SCO UnixWare
- Tru64
- eBD
- Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7 y Windows Server (2000, 2003 y 2008).
- OpenVMS

Características a partir de la versión 5.x

- Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
- Soporte a multiplataforma.
- Procedimientos almacenados.
- Disparadores (*triggers*).
- Cursores.
- Vistas actualizables.
- Soporte a VARCHAR.
- INFORMATION_SCHEMA.

- Modo Strict.
- Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle.
- Motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial).
- Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación (savepoints) con InnoDB.
- Soporte para SSL.
- Query caching.
- Sub-SELECTs (o SELECTs anidados).
- Réplica con un maestro por esclavo, varios esclavos por maestro, sin soporte automático para múltiples maestros por esclavo.
- Indexing y búsqueda de campos de texto completos usando el motor de almacenamiento MyISAM.
- Embedded database library.
- Soporte completo para Unicode.
- Conforme a las reglas ACID usando los motores InnoDB, BDB y Cluster.
- Shared-nothing clustering through MySQL Cluster, etc.

Usuarios destacados

- Amazon.com
- Cox Communications - La cuarta televisión por cable más importante de EEUU, tienen más de 3.600 tablas y aproximadamente dos millones de inserciones cada hora.
- Craigslist
- CNET Networks
- Digg - Sitio de noticias.
- Flickr, usa MySQL para gestionar millones de fotos y usuarios.
- Google - Para el motor de búsqueda de la aplicación AdWords.
- Joomla!, con millones de usuarios.
- phpBB, Uno de los más famosos sitios de foros, con miles de instalaciones y con millones de usuarios.
- LiveJournal - Cerca de 300 millones de páginas servidas cada día.
- NASA
- NetQOS, usa MySQL para la gestión de algunas de las redes más grandes del mundo como las de Chevron, American Express y Boeing.
- Nokia, usa un cluster MySQL para mantener información en tiempo real sobre usuarios de redes de móviles.
- Omniture
- Sabre, y su sistema de reserva de viajes Travelocity.
- Slashdot - con cerca de 50 millones de páginas servidas cada día.
- Wikipedia, sirve más de 200 millones de consultas y 1,2 millones de actualizaciones cada día, con picos de 11.000 consultas por segundo.
- WordPress, con cientos de blogs alojados en él.
- Yahoo! - para muchas aplicaciones críticas.

❖ *Postgresql*

PostgreSQL es un sistema de gestión de bases de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

Entre sus características principales se encuentran las siguientes:

- ***Alta concurrencia***

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*.

- ***Amplia variedad de tipos nativos***

PostgreSQL provee nativamente soporte para: Números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP (IPv4 e IPv6), bloques de direcciones estilo CIDR, direcciones MAC, arrays. Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indizables gracias a la infraestructura GiST de PostgreSQL.

Otras características

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (*foreign keys*).
- Disparadores (*triggers*): Un disparador se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Los disparadores (triggers) son funciones enlazadas a operaciones sobre los datos.

Algunos de los lenguajes que se pueden usar son los siguientes: Un lenguaje propio llamado PL/PgSQL (similar al PL/SQL de oracle), C, C++, Gambas, Java PL/Java web, , PL/Perl, PL/PHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme, Lenguaje para aplicaciones estadísticas R por medio de PL/R.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés). Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés).

1.1.5 USUARIOS Y ADMINISTRADORES DE LAS BASES DE DATOS

Las personas que trabajan con una base de datos se pueden catalogar como usuarios de bases de datos, o bien, como administradores de bases de datos.

Usuarios de bases de datos

Usuarios normales.- Son usuarios no sofisticados que interactúan con el sistema mediante la invocación de alguno de los programas de aplicación permanentes que se ha escrito previamente.

La interfaz de usuario normal para los usuarios normales es una interfaz de formularios, donde el usuario puede rellenar los campos apropiados del formulario o simplemente leer informes generados de la base de datos.

Programadores de aplicaciones.- Son profesionales informáticos que escriben programas de aplicación y pueden elegir entre muchas herramientas para desarrollar interfaces de usuario. Las herramientas de desarrollo rápido de aplicaciones (DRA) permiten al programador de aplicaciones construir formularios e informes sin escribir un programa. Hay también tipos especiales de lenguajes de programación que combinan estructuras de control imperativo (por ejemplo, para bucles for, bucles while e instrucciones if-then-else) con instrucciones del lenguaje de manipulación de datos. Estos lenguajes, llamados lenguajes de cuarta generación, a menudo incluyen características especiales para facilitar la generación de formularios y la presentación de datos en pantalla. La mayoría de los sistemas de bases de datos comerciales incluyen un lenguaje de cuarta generación.

Usuarios sofisticados.- Estos usuarios interactúan con el sistema sin programas escritos, en su lugar, ellos forman sus consultas en un lenguaje de consulta de bases de datos. Cada una de estas consultas se envía al procesador de consultas, cuya función es transformar instrucciones LMD a instrucciones que el gestor de almacenamiento entienda. Los analistas que envían las consultas para explorar los datos en la base de datos entran en esta categoría. Las herramientas de procesamiento analítico en línea (OLAP, Online Analytical Processing) simplifican la labor de los analistas permitiéndoles ver resúmenes de datos de

formas diferentes. Asimismo, permiten al analista seleccionar regiones específicas, examinar los datos con más detalle o examinar los datos con menos detalle.

Otra clase de herramientas para los analistas son las herramientas de recopilación de datos, que les ayudan a encontrar ciertas clases de patrones de datos.

Usuarios especializados.- Son usuarios sofisticados que escriben aplicaciones de bases de datos especializadas que no son adecuadas en el marco de procesamiento de datos tradicional. Entre estas aplicaciones están los sistemas de diseño asistido por computadora, sistemas de bases de conocimientos y sistemas expertos, sistemas que almacenan los datos con tipos de datos complejos (por ejemplo, datos gráficos y datos de audio) y sistemas de modelado del entorno.

Administrador de la base de datos

La persona que tiene el control central (tanto de los datos como de los programas que acceden a esos datos) sobre el sistema se llama administrador de la base de datos (ABD). Las funciones del ABD incluyen las siguientes:

- **Definición del esquema.** El ABD crea el esquema original de la base de datos escribiendo un conjunto de instrucciones de definición de datos en el LDD.
- **Definición de la estructura y del método de acceso.**
- **Modificación del esquema y de la organización física.** Los ABD realizan cambios en el esquema y en la organización física para reflejar las necesidades cambiantes de la organización, o para alterar la organización física para mejorar el rendimiento.
- **Concesión de autorización para el acceso a los datos.** La concesión de diferentes tipos de autorización permite al administrador de la base de datos determinar a qué partes de la base de datos puede acceder cada usuario. La información de autorización se mantiene en una estructura del sistema especial que el sistema de base de datos consulta cuando se intenta el acceso a los datos en el sistema.
- **Mantenimiento rutinario.** Algunos ejemplos de actividades rutinarias de mantenimiento del administrador de la base de datos son:
 - Copia de seguridad periódica de la base de datos, ya sea sobre cinta o en servidores remotos, para prevenir la pérdida de datos en caso de desastres como inundaciones.
 - Asegurarse de que haya suficiente espacio libre en disco para las operaciones normales y aumentarlo si es necesario.
 - Supervisión de los trabajos que se ejecuten en la base de datos y asegurarse de que el rendimiento no se degrada por tareas muy costosas iniciadas por algunos usuarios.

1.1.6 INTERFACES DEL SGBD

Interfaces para navegación basados en menús

Estas interfaces presentan al usuario listas de opciones denominadas menús, que guían al usuario para formar una solicitud. Los menús hacen innecesario memorizar las instrucciones y la sintaxis específica de un lenguaje de consulta, ya que permiten confeccionar la solicitud paso a paso mediante la elección de las diferentes opciones del menú que le va presentando el sistema. Los menús desplegados son una técnica cada vez más popular en las interfaces de usuario basadas en ventanas. Se utilizan a menudo en interfaces de navegación, que permiten al usuario ojear los contenidos de una base de datos usando una forma de exploración no estructurada.

Interfaces basadas en formularios

Las interfaces basadas en formularios presentan un formulario a cada usuario. Éste puede entonces rellenar todas las entradas para insertar nuevos datos o bien rellenar sólo algunos de ellos, en cuyo caso el SGBD obtendrá los datos que coincidan con los que se ha especificado. Los formularios suelen diseñarse y programarse para los usuarios normales como interfaces de transacciones programadas. Muchos SGBD cuentan con lenguajes especiales, los lenguajes de especificación de formularios, con los que los programadores pueden especificar dichos formularios. Algunos sistemas cuentan con utilidades que definen formularios y permiten que el usuario construya interactivamente en pantalla un formulario de muestra.

Interfaces gráficas de usuario

Las interfaces gráficas de usuario (GUI) suelen presentar al usuario los esquemas en forma de diagrama, y éste puede entonces especificar una consulta manipulando el diagrama. En muchos casos, las GUI se combinan con menús y formularios. Casi todas estas interfaces utilizan un dispositivo apuntador, como el ratón, para escoger ciertas partes del diagrama de esquema que se exhibe.

Interfaces de lenguaje natural

Estas interfaces aceptan solicitudes escritas en inglés o en algún otro idioma e intentan entenderlas. Las interfaces de lenguaje natural suelen tener su propio esquema, similar al esquema conceptual de la base de datos. La interfaz consulta las palabras de su esquema y también un conjunto de palabras estándar, para interpretar la solicitud. Si la interpretación tiene éxito, la interfaz genera una consulta de alto nivel que corresponde a la solicitud en lenguaje natural y la envía al SGBD para su procesamiento; en caso contrario, se inicia un diálogo con el usuario para esclarecer la solicitud.

Interfaces para usuarios paramétricos

Los usuarios paramétricos a menudo tienen un conjunto pequeño de operaciones que deben realizar repetidamente. Los analistas de sistemas y los programadores diseñan e implementan una interfaz especial para una clase conocida de usuarios normales. Casi siempre se incluye un conjunto reducido de instrucciones abreviadas, con el fin de reducir al mínimo el número de pulsaciones requeridas para cada solicitud.

Interfaces para el ABD

En su mayoría, los sistemas de base de datos contienen instrucciones privilegiadas que sólo el personal del ABD puede utilizar. Entre ellas están las instrucciones para crear cuentas, establecer los parámetros del sistema, otorgar autorizaciones a las cuentas, modificar los esquemas y reorganizar la estructura de almacenamiento de una base de datos.

Interfaces web para bases de datos

La World Wide Web (Web) es un sistema distribuido de información basado en hipertexto. En los últimos años la Web se ha transformado en la interfaz de usuario con las bases de datos más utilizada. La Web se ha vuelto importante como frontal para las bases de datos por varios motivos: los navegadores Web ofrecen un frontal universal para la información facilitada por los dorsales ubicados en cualquier parte del mundo. El frontal puede ejecutarse en cualquier sistema informático y no es necesario descargar software específico para tener acceso a la información.

Con el crecimiento de los servicios de información y del comercio electrónico en la Web, las bases de datos empleadas para los servicios de información, la ayuda a la toma de decisiones y el procesamiento de transacciones deben estar conectados a la Web. La interfaz de los formularios HTML resulta conveniente para el procesamiento de las transacciones. El usuario puede rellenar los detalles de un formulario de pedido y pulsar un botón de envío para remitir un mensaje al servidor, el cual ejecuta un programa de aplicación correspondiente al formulario de pedido y esta acción, a su vez, ejecutará las transacciones en una base de datos en el sitio del servidor. El servidor da formato a los resultados de la transacción y se los devuelve al usuario.

Otro motivo para el uso de interfaces entre las bases de datos y la Web es que la presentación exclusiva de documentos estáticos (fijos) en un sitio Web presenta algunas limitaciones, aunque el usuario no realice ninguna consulta ni procese ninguna transacción:

- Los documentos Web fijos no permiten que lo que se muestre se adapte al usuario.
- Cuando se actualizan los datos de la compañía, los documentos de la Web se quedan obsoletos si no se actualizan de manera simultánea. El problema se hace más grave si varios documentos de la Web replican datos importantes y hay que actualizarlos todos.

Estos problemas pueden resolverse mediante la generación dinámica de los documentos de la Web a partir de una base de datos. Cuando se solicita un documento se ejecuta un programa en el sitio del servidor, que a su vez ejecuta consultas en una base de datos y genera el documento solicitado de acuerdo con los resultados de las consultas. Siempre que se actualizan datos importantes de la base de datos los documentos generados se actualizan de manera automática. El documento generado también puede personalizarse para el usuario de acuerdo con la información del usuario guardada en la base de datos. Las interfaces Web ofrecen ventajas atrayentes incluso para aplicaciones de bases de datos que sólo se utilizan dentro de una organización. La norma *lenguaje de marcas de hipertexto (HyperText Markup Language, HTML)* permite que el texto reciba formato de manera ordenada y que se destaque la información importante. Los hipervínculos, que son enlaces con otros documentos, pueden asociarse con regiones de los datos mostrados. Al pulsar en

un hipervínculo se captura el documento vinculado y se muestra. Los hipervínculos resultan muy útiles para explorar datos, lo que permite a los usuarios obtener más detalles de las partes de los datos que prefieran.

Finalmente, los navegadores de hoy en día pueden capturar programas junto con los documentos HTML, y ejecutarlos en el navegador, en modo seguro, es decir, sin dañar los datos de la computadora del usuario. Los programas pueden escribirse en lenguajes de guiones del lado del cliente, como Javascript, o ser applets escritas en el lenguaje Java. Estos programas permiten la creación de interfaces de usuario sofisticadas, más de lo que es posible únicamente con HTML y además pueden utilizarse sin descargar ni instalar ningún software por lo que las interfaces web resultan potentes y atractivas visualmente.

❖ *Los localizadores uniformes de recursos*

Un localizador uniforme de recursos (Uniform Resource Locator, *URL*) es un nombre globalmente único para cada documento al que se puede tener acceso en la Web. Un ejemplo de URL es el siguiente:

`http://www.ingenieria.unam.mx/carreras/computacion.html`

La primera parte del URL indica el modo en que se puede tener acceso al documento: *http* indica que se puede tener acceso al documento mediante el protocolo de transferencia de hipertexto (HyperText Transfer Protocol, HTTP), que es un protocolo para transferir documentos HTML. La segunda parte da el nombre único de una máquina que tiene un servidor Web. El resto del URL es el nombre del camino hasta el archivo en la máquina, u otro identificador único del documento dentro de la máquina. Muchos datos de la Web se generan de manera dinámica. Un URL puede contener el identificador de un programa ubicado en la máquina servidora Web, así como argumentos que hay que darle al programa. Un ejemplo de URL de este tipo es

`http://www.google.com/search?q=UNAM`

que dice que el programa search en el servidor `www.google.com` se debe ejecutar con el argumento `q=UNAM`. El programa se ejecuta, utilizando los argumentos dados, y devuelve un documento HTML, que se envía al frontal.

❖ *Los servidores Web*

Los servidores Web son programas que se ejecutan en la máquina servidora, que aceptan las solicitudes de los exploradores Web y devuelven los resultados en forma de documentos HTML. El explorador y el servidor Web se comunican mediante un protocolo denominado protocolo de transferencia de hipertexto (HyperText Transfer Protocol, HTTP), el cual proporciona características muy potentes.

Apache es el servidor web por excelencia. Ha sido uno de los mayores éxitos del software libre y su supremacía entre los servidores de web no se ve amenazada por ningún otro servidor web.

APACHE

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

Características y ventajas

- Apache es, actualmente, uno de los servidores web más utilizados a escala mundial, en gran parte se debe a que es Freeware bajo licencia GNU, pero también se debe a su robustez y sus múltiples posibilidades. Son ya muchas las empresas que gestionan su propio host (ya sea para Intranets o Internet) mediante un servidor Apache, lo cual les permite ahorrar las tasas de hosting privado o público que ofrecen numerosas empresas dedicadas a este tipo de servicios. Con Apache cualquiera puede montar su propio servidor Web y hacer uso de él.
- El hecho de ser multiplataforma, es decir, que soporte infinidad de sistemas operativos distintos entre sí, es también un gran aliciente para su enorme proliferación. Hay versiones de Apache para los sistemas operativos más usados (Windows, Linux, Unix, Solaris, Mac, etc.).
- Una de las principales características de Apache es su extensibilidad basada en una gran modularidad de su código fuente, lo que han facilitado la aparición de módulos de extensión como PHP el cual evita el uso de cgi-bins por completo, facilitando enormemente la programación de aplicaciones en el lado del servidor, especialmente en el campo de acceso a bases de datos.
- Otra de las características más atractivas de este servidor, es que está añadiendo continuamente nuevas características y mejoras, lo cual nos garantiza un crecimiento futuro.
- Es relativamente fácil de configurar, puesto que solo existe 1 fichero de configuración.
- Posee infinidad de paquetes y módulos que nos permiten trabajar con gran cantidad de lenguajes de programación web, así como intérpretes de SQL y otras funciones.
- Permite transacciones seguras mediante SSL (Secure Socket Layer)
- Contiene soporte para Hosts virtuales

Arquitectura del Servidor Apache

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base.**- Módulo con las funciones básicas del Apache
- **Módulos Multiproceso.**- Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- **Módulos Adicionales.**- Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Todas las funciones básicas de los módulos y sus configuraciones se pueden encontrar en el fichero de configuración *httpd.conf*.

1.1.7 ESTRUCTURA DE UN SISTEMA DE BASES DE DATOS

Los componentes funcionales de un sistema de bases de datos se pueden dividir a grandes rasgos en dos componentes: el gestor de almacenamiento y el procesador de consultas.

Gestor de almacenamiento

Un gestor de almacenamiento es un módulo de programa que proporciona la interfaz entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas al sistema. El gestor de almacenamiento es responsable de la interacción con el gestor de archivos. Los datos en bruto se almacenan en disco usando un sistema de archivos, que está disponible habitualmente en un sistema operativo convencional. El gestor de almacenamiento traduce las diferentes instrucciones LMD a órdenes de un sistema de archivos de bajo nivel. Así, el gestor de almacenamiento es responsable del almacenamiento, recuperación y actualización de los datos en la base de datos.

Los componentes del gestor de almacenamiento incluyen:

- Gestor de autorización e integridad
- Gestor de transacciones.
- Gestor de archivos.
- Gestor de memoria intermedia.

El gestor de almacenamiento implementa varias estructuras de datos como parte de la implementación física del sistema:

- Archivos de datos.
- Diccionario de datos.
- Índices.

Procesador de consultas

El procesador de consultas es importante porque ayuda al sistema de bases de datos a simplificar y facilitar el acceso a los datos. Los componentes del procesador incluyen:

- Intérprete del LDD.
- Compilador del LMD.
- Motor de evaluación de consultas.

En la siguiente figura se muestran estos componentes y sus respectivas conexiones:

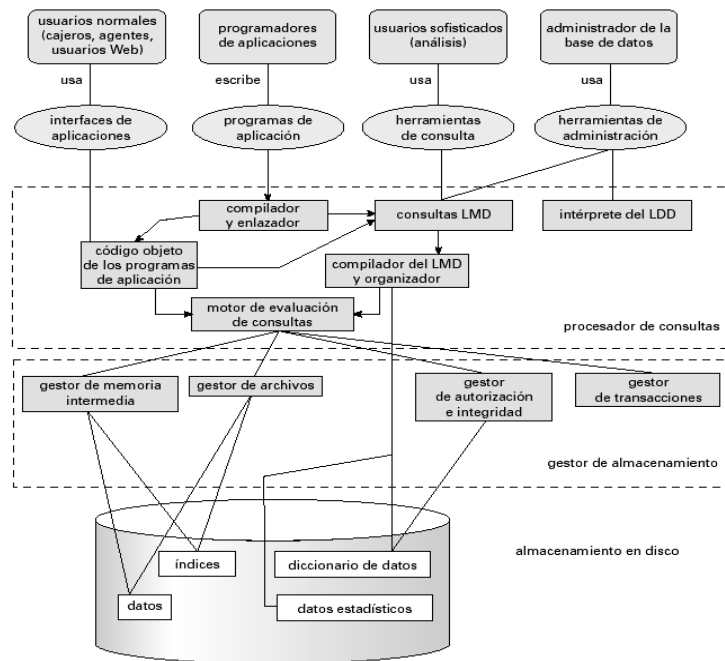


Figura 1.5 Estructura del sistema.

1.1.8 MODELOS DE DATOS

Un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones entre los datos, la semántica de los datos y las restricciones de consistencia.

Los diferentes modelos de datos se clasifican en tres grupos diferentes:

- 1) Modelos lógicos basados en registros
 - Jerárquico
 - Red
 - Relacional
- 2) Modelos lógicos basados en objetos
 - Entidad – Relación
 - Orientado a objetos
- 3) Modelos físicos
 - Unificador
 - Memoria de cuadros o elementos

1) 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. 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.

❖ El Modelo Relacional

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

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional.

Debido a que el modelo relacional está basado en la teoría de conjuntos, los datos se organizan en un tipo especial de conjunto denominado relación, que se define de la siguiente manera: sean los conjuntos D_1, \dots, D_n , denominados dominios, que no tienen por qué ser distintos entre sí. Una relación definida sobre D_1, \dots, D_n es cualquier subconjunto R de D , donde n es el grado o aridad de R . Los dominios son en principio conjuntos finitos de datos. Por tanto, a menos que se indique lo contrario, las relaciones también son finitas. Los elementos de una relación se denominan tuplas. Formalmente, una tupla es:

$$\langle d_1, \dots, d_n \rangle, \text{ donde } d_1 \in D_1, \dots, d_n \in D_n$$

El número de tuplas en una relación es la cardinalidad de la relación. Puesto que una relación es un conjunto, los elementos de este conjunto, las tuplas, han de ser por fuerza distintas. Esto también implica que el orden de las tuplas es irrelevante. El conjunto vacío es una relación particular: la relación nula o vacía. Cada tupla de una relación, junto con el nombre de la relación, representa una aserción (en el sentido lógico).

Las relaciones también pueden ser vistas como tablas, en las que cada tupla es una fila de la tabla. Los nombres de las columnas de la tabla, por otra parte, son los atributos. El conjunto (ordenado) de todos los atributos de una relación R es el esquema de R . Se puede hacer referencia a los atributos de una relación mediante su nombre o por la posición (número de columna) que el atributo ocupa en el esquema de la relación. Las tuplas, por tanto, pueden ser consideradas como matrices de pares (atributo, valor).

El dominio D de R es la colección de valores posibles para un determinado atributo. Cada atributo debe tomar los valores de un dominio subyacente, y sólo uno. Que un elemento sea atómico (también llamado escalar) implica que no puede ser descompuesto en partes más pequeñas.

Un dato, también puede ser compuesto, es decir puede ser descompuesto y hacer uso de los datos atómicos que contiene. En el modelo relacional todos los datos han de ser atómicos, aunque hay una excepción muy particular a esta regla: la relación, que es considerada como un tipo de dato compuesto.

Los términos formales del modelo relacional a menudo son sustituidos por otros de uso más común, debido a que estos términos son muy abstractos para ser usados en la práctica.

Término relacional	Equivalente
Relación	Tabla
Tupla	Fila o registro
Cardinalidad	Número de filas o registros
Atributo	Columna o campo
Grado	Número de columnas o campos
Clave primaria	Identificador único
Dominio	Conjunto de valores legales

Tabla 1.8 Términos relacionales y sus equivalentes.

	Nombre-cliente	Calle-cliente	Estado-cliente
	Juan	Pino Suárez	Veracruz
	Aarón	Nubes	Ciudad de México
	Alejandra	Abasolo	Oaxaca
	Crystal	Ecab	Quintana Roo
	Rosa	Morelos	Estado de México

Registros, Renglones, Tuplas →

↑ Campos, Columnas, Atributos

Tabla 1.9 Ejemplo: La Relación cliente.

Reglas de Codd

Regla 0.- Cualquier DBMS que proclame ser relacional, deberá manejar, completamente, las bases de datos por medio de sus capacidades relacionales.

Regla 1 (Regla de la Información).- Toda la información dentro de una base de datos relacional se representa de manera explícita a nivel lógico y exactamente de una sola manera, como valores en una tabla.

Regla 2 (Regla del acceso garantizado).- Se garantiza que todos y cada uno de los datos (valor atómico) en una base de datos relacional pueden ser leídos recurriendo a una combinación de nombre de la tabla, valor de la llave primaria y nombre de la columna.

Regla 3 (Tratamiento sistemático de valores nulos).- En un DBMS totalmente relacional se soportan los valores nulos (que son distintos de una cadena de caracteres vacía o de una cadena con caracteres en blanco o de cero o cualquier otro número). Para representar información faltante o no aplicable de una forma consistente independientemente del tipo de dato.

Regla 4 (Diccionario dinámico en línea basado en el modelo relacional).- La descripción de la base de datos se representa en el nivel lógico de la misma forma que los datos ordinarios, de modo que los usuarios autorizados puedan aplicar el mismo lenguaje relacional para consultarla, que aquel que emplean con sus datos normales.

Regla 5 (Regla del sublenguaje de datos completo).- Un sistema relacional debe soportar varios lenguajes y varios modos de uso de terminal. Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante una sintaxis bien definida, como cadenas de caracteres y que sea completo, es decir, deberá contener un sublenguaje de datos completo llamado SQL (Structured Query Lenguaje) que comprenda como estándar lo siguiente:

- Definición de datos
- Definición de vistas
- Manipulación de datos
- Restricciones de integridad (manejo)
- Autorización
- Inicio y fin de una transacción

Regla 6 (Regla de actualización de vistas).- Todas las vistas que teóricamente sean actualizables, deben ser actualizadas por medio del sistema.

Regla 7 (Inserción, actualización y borrado de alto nivel).- La posibilidad de manejar una relación base o una relación derivada como un solo operador se aplica a la lectura, inserción, modificación y eliminación de datos.

Regla 8 (Independencia física de datos).- Los programas de aplicación y la actividad en terminales no deberán ser afectados a nivel lógico por cambios en el almacenamiento físico de los datos o en los métodos de acceso.

Regla 9 (Independencia lógica de datos).- Los programas de aplicación y la actividad en terminales no deberán ser afectados por cambios de cualquier tipo, que preserven la información y que teóricamente permitan la afectación, en las tablas base.

Regla 10 (Independencia de integridad).- Las restricciones de integridad de una base de datos relacional deberán poder definirse con el mismo sublenguaje de datos relacional y deberá almacenarse en el catálogo, no en los programas de aplicación.

Regla 11 (Independencia de distribución).- Un DBMS relacional tiene independencia de distribución, esto es, si se manipulan varios fragmentos de la misma base en diferentes medios de almacenamiento, tener la capacidad como si fuera una sola

Regla 12 (Regla de la no subversión).- Si un sistema relacional tiene un lenguaje de bajo nivel (que opere un registro cada vez), ese lenguaje no deberá poder emplearse para subvertir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de más alto nivel.

Normalización

La normalización es un proceso de descomposición sin pérdida, para lograr que nuestras bases de datos estén lo más óptimas posibles.

Una dependencia funcional se representa cuando los valores de un conjunto de atributos de una tupla determinan de manera única los valores de otro conjunto de atributos.

La teoría de normalización está basada en la observación de que cierto conjunto de relaciones presenta mejores propiedades en un medio de actualización, inserción y supresión, que las que presentan otros conjuntos de relaciones que contienen los mismos datos.

La razón de usar el procedimiento de normalización es asegurar que el modelo conceptual de la base de datos funcionará. Esto no significa que una estructura no normalizada no funcionará, sino que puede causar algunos problemas cuando los programadores de aplicación traten de modificar la base de datos.

Las formas normales son una serie de restricciones que se definen sobre las estructuras relacionales para evitar, como ya se señaló, anomalías al efectuar adiciones, eliminaciones o actualizaciones de tuplas. Con el fin de conseguir que una relación cumpla con una forma normal se efectúa un proceso de descomposición. Ésta implica dividir los atributos de una relación en dos subconjuntos (posiblemente con una intersección no vacía) sin que por ello se pierda alguna información contenida en la relación original.

Las formas de normalización fueron propuestas originalmente por Codd, en 1971 y 1972. Posteriormente varios investigadores continuaron trabajando en esta teoría y a lo largo del tiempo han surgido varias formas de normalización que complementan y refuerzan a las enunciadas por Codd.

- **Primera Forma Normal (1FN)**

Una relación normalizada es una relación que tiene sólo valores elementales (o simples) en la intersección de cada renglón y columna. Así, una relación normalizada no tiene grupos repetitivos.

Para normalizar una relación que contiene un sólo grupo repetitivo, se elimina el grupo repetitivo y se forman dos nuevas relaciones. Una relación esta en primera forma normal si no contiene grupos repetitivos.

Una entidad R está en Primera Forma Normal (1FN) si los valores, para cada atributo A ∈ R, son atómicos.

Esto implica, que los valores en el dominio no deberán ser listas o conjuntos de valores. Las actualizaciones representan un problema potencial si la entidad no está en 1NF.

- ***Segunda Forma Normal (2FN)***

Para eliminar anomalías de la primera forma normal, se deben eliminar las dependencias parciales. Una relación está en segunda forma normal, si está en primera forma normal y se han eliminado las dependencias parciales.

Se entiende por dependencia parcial cuando los atributos no llave dependen sólo de una parte de una llave compuesta.

El segundo paso de la normalización es establecer las claves y relacionarlas con los campos de datos. En la primera forma normalizada, el renglón entero de la tabla (tupla) depende de todos los campos de claves. En la segunda forma normalizada, se hace un intento de establecer los campos de datos que están relacionados con alguna parte de la clave completa. Si los campos de datos sólo dependen de una parte de la clave, la clave y los campos conectados a la clave parcial son susceptibles de separarse en registros independientes. La división de la primera tabla normalizada, en una serie de tablas en las que cada campo sólo depende de la clave completa se llama la segunda forma normalizada.

- ***Tercera Forma Normal (3FN)***

Una relación está en tercera forma normal si está en segunda forma normal y no tiene dependencias transitivas.

Esto es, cada atributo no llave depende totalmente de la llave primaria y no hay dependencias transitivas.

Una dependencia transitiva ocurre cuando un atributo no llave depende de uno o más atributos no llave.

- ***Forma Normal de Boyce-Codd (FNBC)***

La tabla se encuentra en BCNF si cada determinante, atributo que determina completamente a otro, es clave candidata.

- ***Cuarta Forma Normal (4FN)***

Una entidad está en cuarta forma normal cuando:

- Está en tercera forma normal y sus atributos no solamente dependen de la llave primaria o compuesta, sino que también del valor de la llave (dependencia del valor).
- Un atributo cuya existencia es opcional se reubica en una entidad donde este atributo es obligatorio (dependencia de la existencia).

En otras palabras, una tabla se encuentra en 4FN si, y sólo si, para cada una de sus dependencias múltiples no funcionales $X \twoheadrightarrow Y$, siendo X una super clave que, X es o una clave candidata o un conjunto de claves primarias.

- **Quinta Forma Normal (5FN)**

Una tabla se encuentra en 5FN si:

- La tabla esta en 4FN.
- No existen relaciones de dependencias no triviales que no siguen los criterios de las claves. Una tabla que se encuentra en la 4FN se dice que está en la 5FN si, y sólo si, cada relación de dependencia se encuentra definida por las claves candidatas.

Desnormalización

Las relaciones normalizadas evitan anomalías durante la modificación, es por eso que son preferibles que aquellas relaciones que no están normalizadas. Sin embargo existen ocasiones en las que esta normalización no es lo más conveniente o no merece la pena realizarla.

2) Modelos lógicos basados en objetos

Se usan para describir datos en el nivel conceptual y en el de visión, es decir, con estos modelos representamos los datos tal y como los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente.

- ❖ **Modelo Entidad – Relación (E – R)**

El modelo entidad-relación (E-R) es un modelo de datos de alto nivel. Está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre estos objetos. El modelo E-R es un modelo de datos semántico cuyo aspecto yace en la representación del significado de los datos. Hay tres nociones básicas que emplea este modelo: conjuntos de entidades, conjuntos de relaciones y atributos.

Características del modelo E – R extendido

Aunque los conceptos básicos de E-R pueden modelar la mayoría de las características de las bases de datos, algunos aspectos de una base de datos pueden ser más adecuadamente expresados mediante ciertas extensiones del modelo E-R básico tales como: la especialización, la generalización, los conjuntos de entidades de nivel más alto y más bajo, la herencia de atributos y la agregación.

❖ **Modelo Orientado a Objetos**

El modelo orientado a objetos define una base de datos en términos de objetos, sus propiedades y sus operaciones. Se basa en el concepto de encapsulamiento de datos y código que opera sobre éstos en un objeto. Los objetos con la misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías o grafos acíclicos. El conjunto de clases está estructurado en sub y superclases basado en una extensión del concepto ISA del modelo Entidad - Relación. Puesto que el valor de un dato en un objeto también es un objeto, es posible representar el contenido del objeto dando como resultado un objeto compuesto. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos denominados métodos.

3) Modelos físicos de datos

También llamados “Modelos de datos primitivos”, describen los datos en el nivel más bajo y permiten identificar algunos detalles de implantación para el manejo del hardware de almacenamiento. Son muy poco usados. Entre los cuales podemos encontrar el modelo unificador y el de memoria de cuadros o elementos.

1.2 REDES DE COMPUTADORAS

Una red de computadoras es un conjunto de máquinas (computadoras) interconectadas entre sí mediante uno o más medios de transmisión a fin de llevar a cabo la transferencia eficiente y confiable de información, recursos y servicios. Esta interconexión puede ser a través de un enlace físico (alambrado) o inalámbrico.

1.2.1 TIPOS DE REDES

Las bases de datos distribuidas y los sistemas cliente-servidor se construyen en torno a las redes de comunicación. Existen básicamente dos clases de redes: las redes de área local y las redes de área amplia. La diferencia principal entre ambas es la forma en que están distribuidas geográficamente y esto implica importantes variaciones en la velocidad y en la fiabilidad de la red de comunicación y quedan reflejadas en el diseño del sistema operativo distribuido.

❖ **Redes de área local**

Una LAN (Local Area Network) es una red de comunicaciones que interconecta varios dispositivos y proporciona un medio para el intercambio de información entre ellos. Su cobertura es pequeña ya que su extensión está limitada físicamente a un edificio o a un conjunto de edificios próximos hasta una distancia de 100 metros

Los enlaces más comunes en una red de área local son el par trenzado, el cable coaxial, la fibra óptica y, cada vez más, las conexiones inalámbricas. La velocidad de comunicación varía entre unos pocos megabits por segundo (en las redes de área local inalámbricas), a un gigabit por segundo para Gigabit Ethernet. La Ethernet estándar funciona a 10 megabits por segundo, mientras que Fast Ethernet llega a 100 megabits por segundo.

Para las LAN hay muy diversas configuraciones. De entre ellas, las más habituales son las LAN conmutadas y las LAN inalámbricas. Dentro de las conmutadas, las más populares son las LAN Ethernet, constituidas por un único conmutador, o alternativamente, implementadas mediante un conjunto de conmutadores interconectados entre sí. Otro ejemplo muy relevante son las LAN ATM, caracterizadas por utilizar tecnología de red ATM en un entorno local. Por último, son también destacables las LAN con canal de fibra óptica (Fiber Channel). En las LAN inalámbricas se utilizan diversos tipos de tecnologías de transmisión y distintos tipos de configuraciones.

❖ Redes de área amplia

Las redes de área amplia (WAN, Wide Area Networks) surgen a finales de los 60 principalmente como un proyecto de investigación académica para proporcionar una comunicación eficiente entre varios lugares permitiendo que una gran comunidad de usuarios pudiera compartir hardware y software de una manera conveniente y económica. Generalmente, se considera como redes de área amplia a todas aquellas que cubren una extensa área geográfica, requieren atravesar rutas de acceso público y utilizan, al menos parcialmente, circuitos proporcionados por una entidad proveedora de servicios de telecomunicación. Una WAN consiste en una serie de dispositivos de conmutación interconectados. La transmisión generada por cualquier dispositivo se encaminará a través de estos nodos internos hasta alcanzar el destino. A estos nodos internos no les concierne el contenido de los datos, al contrario, su función es proporcionar el servicio de conmutación, necesario para transmitir los datos de nodo en nodo hasta alcanzar su destino final.

A principios de los años 60 se desarrollaron sistemas que permitían que terminales remotos se conectaran a una computadora central a través de la línea telefónica, pero no eran verdaderas WANs. *Arpanet* fue la primera WAN que se diseñó y se desarrolló. El trabajo en Arpanet comenzó en 1968. Arpanet ha crecido de tal forma que ha pasado de ser una red experimental de cuatro puestos a una red de redes extendida por todo el mundo, Internet, abarcando a cientos de millones de sistemas de computación. Los enlaces típicos de Internet son las líneas de fibra óptica y, a veces, los canales vía satélite. Las transferencias de datos para los enlaces de área amplia varían normalmente entre los pocos megabits por segundo y cientos de gigabits por segundo. El último enlace, hasta el puesto del usuario, se basa a menudo en la tecnología de bucle de suscriptor digital (DSL, digital subscriber loop) (que soporta unos pocos megabits por segundo), o módem de cable (que soporta 10 megabits por segundo) o conexiones de módem telefónico sobre líneas telefónicas (que soportan hasta 56 kilobits por segundo).

Pueden distinguirse dos tipos de WANs:

1. En las WAN de conexión discontinua, como las basadas en conexiones por radio, las computadoras están conectadas a la red sólo durante intervalos del tiempo.
2. En las WAN de conexión continua, como Internet, las computadoras están conectadas a la red continuamente.

Tradicionalmente, las WAN se han implementado usando una de las dos tecnologías siguientes: conmutación de circuitos y conmutación de paquetes. Últimamente, se está

empleando como solución la técnica de retransmisión de tramas (frame relay), así como las redes ATM.

Cabe mencionar que también existen las redes MAN (Metropolitan Area Network), CAN (Campus Area Network), PAN (Personal Area Network), GAN (Global Area Network), entre otras.

1.2.2 ARQUITECTURA DE PROTOCOLOS

Una arquitectura de protocolos es una estructura en capas de elementos hardware y software que facilita el intercambio de datos entre sistemas y posibilita aplicaciones distribuidas, como el comercio electrónico y la transferencia de archivos.

En los sistemas de comunicación, en cada una de las capas de la arquitectura de protocolos se implementa uno o más protocolos comunes. Cada protocolo proporciona un conjunto de reglas para el intercambio de datos entre sistemas.

❖ Modelo de referencia OSI

El Modelo de Referencia de Interconexión de Sistemas Abiertos, conocido mundialmente como Modelo OSI (Open System Interconnection), fue creado en 1977 por la Organización Internacional de Estandarización (ISO, International Organization for Standardization) y en él pueden modelarse o referenciarse diversos dispositivos que reglamenta la Unión Internacional de Telecomunicaciones (ITU, International Telecommunication Union), con el fin de poner orden entre todos los sistemas y componentes requeridos en la transmisión de datos, además de simplificar la interrelación entre fabricantes.

Los estándares son necesarios para promover la interoperabilidad entre los equipos de distintos fabricantes, así como para facilitar economías de gran escala. Debido a la complejidad que implican las comunicaciones, un solo estándar no es suficiente. OSI nace de esta necesidad de uniformizar los elementos que participan en la solución del problema de comunicación entre equipos de cómputo de diferentes fabricantes.

El modelo OSI tiene siete capas. La función de cada una de ellas consiste en proporcionar servicios a las capas superiores.

7. APLICACIÓN
6. PRESENTACIÓN
5. SESIÓN
4. TRANSPORTE
3. RED
2. ENLACE DE DATOS
1. FÍSICA

Tabla 1.10 Las capas de OSI.

Capa física

La capa física se encarga de la interfaz física entre los dispositivos. Además, define las reglas que rigen en la transmisión de los bits. La capa física tiene cuatro características importantes:

- Mecánicas.- relacionadas con las propiedades físicas de la interfaz con el medio de transmisión. Normalmente, dentro de estas características se incluye la especificación del conector que transmite las señales a través de conductores. A estos últimos se les denominan circuitos.
- Eléctricas.- especifican cómo se representan los bits así como su velocidad de transmisión.
- Funcionales.- especifican las funciones que realiza cada uno de los circuitos de la interfaz física entre el sistema y el medio de transmisión.
- De procedimiento.- especifican la secuencia de eventos que se llevan a cabo en el intercambio del flujo de bits a través del medio físico.

Algunos ejemplos de estándares de esta capa son el EIA-232-F y algunas secciones de los estándares de comunicaciones inalámbricas y LAN.

Capa de enlace de datos

La tarea primordial de la capa de enlace de datos consiste en, a partir de un medio de transmisión común y corriente, transformarlo en una línea sin errores de transmisión para la capa de red. Esta tarea la realiza al hacer que el emisor trocee la entrada de datos en tramas de datos (típicamente constituidas por algunos cientos de octetos), y las transmita en forma secuencial y procese las tramas de asentimiento, devueltas por el receptor.

Mientras que la capa física proporciona exclusivamente un servicio de transmisión de datos, la capa de enlace de datos intenta hacer que el enlace físico sea fiable. Además proporciona los medios para activar, mantener y desactivar el enlace. El principal servicio proporcionado por esta capa a las capas superiores es el de detección y control de errores.

Ejemplos de estándares de esta capa son HDLC, LLC, Ethernet, Token Ring, PPP, Frame Relay, RDSI, ATM, IEEE 802.11, FDDI.

Capa de red

La capa de red se ocupa del control de la operación de la subred. Un punto de suma importancia en su diseño, es la determinación sobre cómo encaminar los paquetes del origen al destino.

La capa de red realiza la transferencia de la información entre sistemas finales a través de algún tipo de red de comunicación. Libera a las capas superiores de la necesidad de tener conocimiento sobre la transmisión de datos subyacente y las tecnologías de conmutación utilizadas para conectar los sistemas. En esta capa, el computador establecerá un diálogo con la red para especificar la dirección destino y solicitar ciertos servicios, como por ejemplo, la gestión de prioridades.

Ejemplos de protocolos en esta capa son: IP, ICMP, IGMP, X.25, CLNP, ARP, RARP, BGP, OSPF, RIP, IGRP, EIGRP, IPX y DDP.

Capa de transporte

La función principal de la capa de transporte consiste en aceptar los datos de la capa de sesión, dividirlos, siempre que sea necesario, en unidades más pequeñas, pasarlos a la capa de red y asegurar que todos ellos lleguen correctamente al otro extremo. Además, todo este trabajo se debe hacer de manera eficiente, de tal forma que aisle la capa de sesión de los cambios inevitables a los que está sujeta la tecnología del hardware.

La capa de transporte proporciona un mecanismo para intercambiar datos entre sistemas finales. El servicio de transporte orientado a conexión asegura que los datos se entreguen libres de errores, en orden y sin pérdidas ni duplicaciones. La capa de transporte también puede estar involucrada en la optimización del uso de los servicios de red, y en proporcionar la calidad del servicio solicitada.

El tamaño y la complejidad de un protocolo de transporte dependen de cómo de fiables sean los servicios de red y las redes subyacentes. Entre los protocolos de esta capa se encuentran los siguientes: TCP, UDP, RTP, SCTP y SPX.

Capa de sesión

La capa de sesión permite que los usuarios de diferentes máquinas puedan establecer sesiones entre ellos. Esta capa proporciona los mecanismos para controlar el diálogo entre las aplicaciones de los sistemas finales. En muchos casos, los servicios de la capa de sesión son parcialmente, o incluso, totalmente prescindibles. No obstante, en algunas aplicaciones su utilización es ineludible. La capa de sesión proporciona los siguientes servicios:

- Control del diálogo.- éste puede ser simultáneo en los dos sentidos (full-duplex) o alternado en ambos sentidos (half-duplex).
- Agrupamiento.- el flujo de datos se puede marcar para definir grupos de datos.
- Recuperación.- la capa de sesión puede proporcionar un procedimiento de puntos de comprobación, de forma que si ocurre algún tipo de fallo entre puntos de comprobación, la entidad de sesión puede retransmitir todos los datos desde el último punto de comprobación.

ISO ha definido una normalización para la capa de sesión que incluye como opciones los servicios antes descritos. Entre los protocolos de esta capa, podemos mencionar los siguientes: TLS, SSH, ISO 8327 / CCITT X.225, RPC, NetBIOS, TELNET.

Capa de presentación

La capa de presentación define el formato de los datos que se van a intercambiar entre las aplicaciones y ofrece a los programas de aplicación un conjunto de servicios de transformación de datos. La capa de presentación define la sintaxis utilizada entre las entidades de aplicación y proporciona los medios para seleccionar y modificar la representación utilizada. Algunos ejemplos de servicios específicos que se pueden realizar en esta capa son los de compresión y cifrado de datos. Algunos protocolos de esta capa son los siguientes: XDR, ASN.1, SMB y AFP.

Capa de aplicación

La capa de aplicación proporciona a los programas de aplicación un medio para que accedan al entorno OSI. A esta capa pertenecen las funciones de administración y los mecanismos genéricos necesarios para la implementación de aplicaciones distribuidas. Además, en esta capa también residen las aplicaciones de uso general como, por ejemplo, la transferencia de archivos, el correo electrónico y el acceso desde terminales a computadores remotos, entre otras. Algunos ejemplos son: FTP, TFTP, HTTP, SMTP, DNS, TELNET, SNMP, SSH, SCP, NFS, RTSP, Feed, Webcal, y POP3.

❖ La arquitectura de protocolos TCP/IP

La arquitectura de protocolos TCP/IP es resultado de la investigación y desarrollo llevados a cabo en la red experimental de conmutación de paquetes ARPANET, financiada por la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA, Defense Advanced Research Projects Agency), y se denomina globalmente como la familia de protocolos TC/IP. Esta familia consiste en una extensa colección de protocolos que se han especificado como estándares de Internet por parte de IAB (Internet Architecture Board).

APLICACIÓN
TRANSPORTE
INTERNET
ACCESO A LA RED

Tabla 1.11 Capas de TCP/IP.

Capa de acceso a la red

Esta capa también se denomina capa de host a red. Es la capa que se ocupa de todos los aspectos que requiere un paquete IP para realizar realmente un enlace físico y luego realizar otro enlace físico. Esta capa incluye los detalles de tecnología LAN y WAN y todos los detalles de la capa física y de enlace de datos del modelo OSI. Es decir, define la interfaz física entre el dispositivo de transmisión de datos y el medio de transmisión o red. Se encarga de la especificación de las características del medio de transmisión, la naturaleza de las señales, la velocidad de los datos y cuestiones afines. La capa de acceso a la red es responsable del intercambio de datos entre el sistema final y la red a la cual está conectado. El emisor debe proporcionar a la red la dirección del destino, de tal manera que ésta pueda encaminar los datos hasta el destino apropiado. El emisor puede requerir ciertos servicios que pueden ser proporcionados por el nivel de red. El software en particular que se use en esta capa dependerá del tipo de red que se disponga. Así, se han desarrollado, entre otros, diversos estándares para la conmutación de circuitos, la conmutación de paquetes (por ejemplo, retransmisión de tramas) y para las redes de área local (por ejemplo, Ethernet).

Capa de Internet

El propósito de la capa de Internet es enviar paquetes origen desde cualquier red en la red y que estos paquetes lleguen a su destino independientemente de la ruta y de las redes que

recorrieron para llegar hasta allí. El protocolo específico que rige esta capa se denomina Protocolo Internet (IP). En esta capa se produce la determinación de la mejor ruta y la conmutación de paquetes.

La capa de Internet contiene 5 protocolos: IP, ARP, ICMP, RARP e IGMP.

Capa de transporte

La capa de transporte se refiere a los aspectos de calidad del servicio con respecto a la confiabilidad, el control de flujo y la corrección de errores. Uno de sus protocolos, el protocolo para el control de la transmisión (TCP), ofrece maneras flexibles y de alta calidad para crear comunicaciones de red confiables, sin problemas de flujo y con un nivel de error bajo. TCP es un protocolo orientado a la conexión. Mantiene un diálogo entre el origen y el destino mientras empaqueta la información de la capa de aplicación en unidades denominadas segmentos. Orientado a la conexión no significa que el circuito exista entre los computadores que se están comunicando (esto sería una conmutación de circuito). Significa que los segmentos de capa 4 viajan de un lado a otro entre dos hosts para comprobar que la conexión exista lógicamente para un determinado período. Esto se conoce como conmutación de paquetes.

Los protocolos de las capas anteriores permiten enviar información de un equipo a otro. La capa de transporte permite que las aplicaciones que se ejecutan en equipos remotos puedan comunicarse. El problema es identificar estas aplicaciones. De hecho, según el equipo y su sistema operativo, la aplicación puede ser un programa, una tarea, un proceso, etc. Además, el nombre de la aplicación puede variar de sistema en sistema. Es por ello que se ha implementado un sistema de numeración para poder asociar un tipo de aplicación con un tipo de datos. Estos identificadores se denominan puertos.

La capa de transporte contiene dos protocolos que permiten que dos aplicaciones puedan intercambiar datos independientemente del tipo de red (es decir, independientemente de las capas inferiores). Estos dos protocolos son los siguientes:

- TCP, un protocolo orientado a conexión que brinda detección de errores;
- UDP, un protocolo no orientado a conexión en el que la detección de errores es obsoleta.

Capa de aplicación

Los diseñadores de TCP/IP sintieron que los protocolos de nivel superior deberían incluir los detalles de las capas de sesión y presentación. Simplemente crearon una capa de aplicación que maneja protocolos de alto nivel, aspectos de representación, codificación y control de diálogo. El modelo TCP/IP combina todos los aspectos relacionados con las aplicaciones en una sola capa y garantiza que estos datos estén correctamente empaquetados para la siguiente capa.

Existen diferentes tipos de aplicaciones para esta capa, pero la mayoría son servicios de red o aplicaciones brindadas al usuario para proporcionar la interfaz con el sistema operativo. Se pueden clasificar según los servicios que brindan:

- Servicios de administración de archivos e impresión (transferencia);
- Servicios de conexión a la red;

- Servicios de conexión remota;
- Diversas utilidades de Internet.

El siguiente diagrama ilustra algunos de los protocolos comunes especificados por el modelo de referencia TCP/IP.

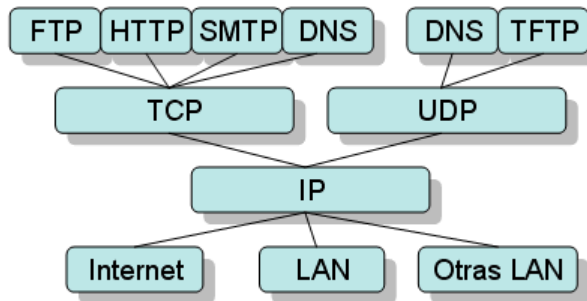


Figura 1.6 Gráfico de protocolo.

En la capa de aplicación, aparecen distintas tareas de red. Estas aplicaciones incluyen las siguientes:

- *FTP*: File Transfer Protocol (Protocolo de transferencia de archivos)
- *HTTP*: Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto)
- *SMTP*: Simple Mail Transfer Protocol (Protocolo de transferencia de correo simple)
- *DNS*: Domain Name System (Sistema de nombres de dominio)
- *TFTP*: Trivial File Transfer Protocol (Protocolo trivial de transferencia de archivo)

El modelo TCP/IP tiene su máxima flexibilidad en la capa de aplicación para los desarrolladores de software. La capa de transporte involucra dos protocolos: el protocolo de control de transmisión (TCP) y el protocolo de datagrama de usuario (UDP). La capa inferior, la capa de acceso de red, se relaciona con la tecnología específica de LAN o WAN que utiliza.

En el modelo TCP/IP existe solamente un protocolo de red: el protocolo Internet, o IP, independientemente de la aplicación que solicita servicios de red o del protocolo de transporte que se utiliza. Esta es una decisión de diseño deliberada. IP sirve como protocolo universal que permite que cualquier computador en cualquier parte del mundo pueda comunicarse en cualquier momento.

Finalmente tenemos que:

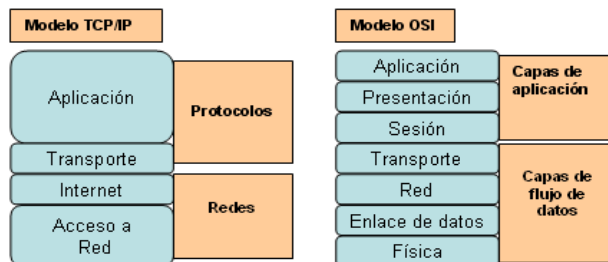


Figura 1.7 Comparación entre las arquitecturas de protocolos TCP/IP y OSI.

1.2.3 DIRECCIONES IP

Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red o nivel 3 del modelo de referencia OSI. Las direcciones IP son únicas e irrepetibles para cada máquina. Para ser precisos, cada dirección es única para cada una de las interfaces de red IP de cada máquina.

Internet es en realidad la interconexión de muchas redes individuales. Cada red debe conocer su propia dirección y la de cualquier otra con la cual quiera comunicarse. Para formar parte de internet, las empresas que brindan los servicios de interconexión necesitan un número de red internet, el cual se debe solicitar al "Network Information Center" (NIC). Este único número es incluido en cualquier paquete que salga de la red en la cual se esté en internet.

El sistema de registro de Internet está estructurado de la siguiente manera. El IANA (Internet Assigned Numbers Authority) tiene bajo su cargo la responsabilidad de administrar la asignación de direcciones IP en el ámbito global, esta organización realiza asignaciones de grandes bloques a los cinco Registros Regionales que existen actualmente los cuales tienen un alcance de influencia continental, en Latinoamérica se encuentra LACNIC, el cual delega la responsabilidad a registros locales como NIC México para administrar el espacio de direcciones a nivel país.

IPv4

IPv4 es la versión 4 del Protocolo IP. Esta fue la primera versión del protocolo que se implementó extensamente, y forma la base de Internet.

Una dirección en IPv4 se representa mediante un número binario de 32 bits (4 bytes), por lo que soporta $2^{32} = 4\,294\,967\,296$ direcciones de red diferentes y únicas. Las direcciones IP se pueden expresar como números de notación decimal: se dividen los 32 bits de la dirección en cuatro octetos. El valor decimal de cada octeto puede ser entre 0 y 255 (el número binario de 8 bits más alto es 11111111 y esos bits, de derecha a izquierda, tienen valores decimales de 1, 2, 4, 8, 16, 32, 64 y 128, lo que suma 255 en total).

En la expresión de direcciones IPv4 en decimal se separa cada octeto por un carácter ".". Cada uno de estos octetos puede estar comprendido entre 0 y 255, salvo algunas excepciones. Los ceros iniciales, si los hubiera, se pueden obviar. Ejemplo de representación de dirección IPv4:

134.13.23.65

Hay tres clases de direcciones IP que una organización puede recibir de parte de la Internet Corporation for Assigned Names and Numbers (ICANN): clase A, clase B y clase C. En la actualidad, ICANN reserva las direcciones de clase A para los gobiernos de todo el mundo (aunque en el pasado se le hayan otorgado a empresas de gran envergadura como, por ejemplo, Hewlett Packard) y las direcciones de clase B para las medianas empresas, cabe mencionar que RedUNAM pertenece a esta clase. Se otorgan direcciones de clase C para entornos mucho más pequeños; la clase D está destinada al tráfico multicast IP, y la clase E,

de momento, no tienen ningún uso concreto más bien es para usos futuros. Cada clase de red permite una cantidad fija de equipos (hosts).

Clase	Primeros Bits	Rango del 1er octeto	Rango de direcciones de red	Rango de direcciones de host	No. de redes	No. de hosts
A	0	1 – 127	1 – 127	0.0.1 – 255.255.254	126	16777214
B	10	128 – 191	128.0 – 191.255	0.1 – 255.254	16384	65534
C	110	192 – 223	192.0.0 – 223.255.255	1 – 254	2097152	254
D	1110	224 – 239	No hay	No hay	No hay	No hay
E	1111	240 – 255	No hay	No hay	No hay	No hay

Tabla 1.12 Clases de redes.

La dirección 0.0.0.0 es utilizada por las máquinas cuando están arrancando o no se les ha asignado dirección.

La dirección que tiene su parte de host a cero sirve para definir la red en la que se ubica. Se denomina dirección de red.

La dirección que tiene su parte de host a unos sirve para comunicar con todos los hosts de la red en la que se ubica. Se denomina dirección de broadcast.

Las direcciones 127.x.x.x se reservan para pruebas de retroalimentación. Se denomina dirección de bucle local o loopback.

Hay ciertas direcciones en cada clase de dirección IP que no están asignadas y que se denominan direcciones privadas. Las direcciones privadas pueden ser utilizadas por los hosts que usan traducción de dirección de red (NAT) para conectarse a una red pública o por los hosts que no se conectan a Internet. En una misma red no pueden existir dos direcciones iguales, pero sí se pueden repetir en dos redes privadas que no tengan conexión entre sí o que se sea a través de NAT. Las direcciones privadas son:

- Clase A: 10.0.0.0 a 10.255.255.255
- Clase B: 172.16.0.0 a 172.31.255.255
- Clase C: 192.168.0.0 a 192.168.255.255

IPv6

IPv6 es la versión 6 del Protocolo de Internet. Fue diseñado por Steve Deering de Xerox PARC y Craig Mudge, IPv6 está destinado a sustituir al estándar IPv4, cuyo límite en el número de direcciones de red admisibles está empezando a restringir el crecimiento de Internet y su uso. Pero el nuevo estándar mejorará el servicio globalmente, proporcionando a futuras celdas telefónicas y dispositivos móviles con sus direcciones propias y permanentes, etc. Al día de hoy se calcula que las dos terceras partes de las direcciones que ofrece IPv4 ya están asignadas. Adoptado por el *Internet Engineering Task Force* en 1994 (cuando era llamado "IP Next Generation" o IPng), IPv6 cuenta con un pequeño porcentaje de las direcciones públicas de Internet, que todavía están dominadas por IPv4. La adopción de IPv6 ha sido frenada por la traducción de direcciones de red (NAT), que alivia parcialmente el problema de la falta de direcciones IP. Pero NAT hace difícil o imposible el

uso de algunas aplicaciones P2P, como son la voz sobre IP (VoIP) y juegos multiusuario. Además, NAT rompe con la idea originaria de Internet donde todos pueden conectarse con todos.

Las características principales de IPv6 son las siguientes:

- Mayor espacio de direcciones.
- “Plug & Play”: Autoconfiguración.
- Seguridad intrínseca en el núcleo del protocolo (IPsec).
- Calidad de Servicio (QoS) y Clase de Servicio (CoS).
- Multicast: Envío de un mismo paquete a un grupo de receptores.
- Anycast: Envío de un paquete a un receptor dentro de un grupo.
- Paquetes IP eficientes y extensibles, sin que haya fragmentación en los encaminadores (routers), alineados a 64 bits (preparados para su proceso óptimo con los nuevos procesadores de 64 bits), y con una cabecera fija, más simple, que agiliza su procesamiento por parte del encaminador (router).
- Posibilidad de paquetes con carga útil (datos) de más de 65 535 bytes.
- Encaminado (enrutado) más eficiente en el troncal (backbone) de la red, debido a una jerarquía de direccionamiento basada en la agregación.
- Renumeración y “multi-homing”, que facilita el cambio de proveedor de servicios.
- Características de movilidad.
- Privacidad.

El número de direcciones IPv6 posibles es de $2^{128} \approx 3.4 \times 10^{38}$. Este número puede también representarse como 16^{32} , con 32 dígitos hexadecimales, cada uno de los cuales puede tomar 16 valores.

En muchas ocasiones las direcciones IPv6 están compuestas por dos partes lógicas: un prefijo de 64 bits y otra parte de 64 bits que corresponde al identificador de interfaz, que casi siempre se genera automáticamente a partir de la dirección MAC de la interfaz a la que está asignada la dirección.

Las direcciones IPv6 son identificadores de 128 bits para interfaces y conjunto de interfaces. Dichas direcciones se clasifican en tres tipos:

- **Unicast:** Identificador para una única interfaz. Un paquete enviado a una dirección unicast es entregado sólo a la interfaz identificada con dicha dirección. Es el equivalente a las direcciones IPv4 actuales.
- **Anycast:** Identificador para un conjunto de interfaces (típicamente pertenecen a diferentes nodos). Un paquete enviado a una dirección anycast es entregado en una (cualquiera) de las interfaces identificadas con dicha dirección (la más próxima, de acuerdo a las medidas de distancia de protocolo de encaminado).
- **Multicast:** Identificador para un conjunto de interfaces (por lo general pertenecientes a diferentes nodos). Un paquete enviado a una dirección multicast es entregado a todas las interfaces identificadas por dicha dirección. La misión de este tipo de paquetes es evidente: aplicaciones de retransmisión múltiple (broadcast).

La representación de las direcciones IPv6 sigue el siguiente esquema:

- a) $x:x:x:x:x:x:x$, donde “x” es un valor hexadecimal de 16 bits, de la proporción correspondiente a la dirección IPv6. No es preciso escribir los ceros a la izquierda de cada campo. Ejemplos:

FEDC:BA98:7654:3210:FEDC:BA98:4567:2311
1080:0:0:0:8:800:300C:417A

- b) Dado que, por el direccionamiento que se ha definido, podrán existir largas cadenas de bits “cero”, se permite la escritura de su abreviación, mediante el uso de “::”, que representa múltiples grupos consecutivos de 16 bits “cero”. Este símbolo sólo puede aparecer una vez en la dirección IPv6. Ejemplos:

Las direcciones:

1080:0:0:0:8:800:300C:417A (una dirección unicast)
FF01:0:0:0:0:0:101 (una dirección multicast)
0:0:0:0:0:0:1 (la dirección loopback)
0:0:0:0:0:0:0 (una dirección no especificada)

Pueden representarse como:

1080::8:800:300C:417A (una dirección unicast)
FF01::101 (una dirección multicast)
::1 (la dirección loopback)
:: (una dirección no especificada)

- c) Una forma alternativa y muy conveniente, cuando nos hallemos en un entorno mixto IPv4 e IPv6, es $x:x:x:x:x:d:d:d:d$, donde “x” representa valores hexadecimales de 16 bits (6 porciones de mayor peso), y “d” representa valores decimales de las 4 porciones de 8 bits de menor peso (representación estándar IPv4). Ejemplos:

0:0:0:0:0:13.1.68.3
0:0:0:0:FFFF:129.144.52.38

Pueden representarse como:

::13.1.68.3
::FFFF:129.144.52.38

La representación de los prefijos IPv6 se realiza del siguiente modo:

Dirección IPv6 / Longitud del prefijo

donde:

- Dirección IPv6 = una dirección IPv6 en cualquiera de las notaciones válidas.

- Longitud del prefijo = valor decimal indicando cuantos bits contiguos de la parte izquierda de la dirección componen el prefijo.

Por ejemplo, las representaciones válidas del prefijo de 60 bits 12AB00000000CD3, son:

```
12AB:0000:0000:CD30:0000:0000:0000:0000 / 60
12AB::CD30:0:0:0:0 / 60
12AB:0:0:CD30:: / 60
```

Por tanto, para escribir una dirección completa, indicando la subred, podríamos hacerlo como:

```
12AB:0:0:CD30:123:4567:89AB:CDEF / 60
```

Los tipos de direcciones IPv6 pueden identificarse tomando en cuenta los primeros bits de cada dirección.

- `::/128` – la dirección con todo ceros se utiliza para indicar la ausencia de dirección, y no se asigna ningún nodo.
- `::1/128` – la dirección de loopback es una dirección que puede usar un nodo para enviarse paquetes a sí mismo (corresponde con 127.0.0.1 de IPv4). No puede asignarse a ninguna interfaz física.
- `::/96` – La dirección IPv4 compatible se usa como un mecanismo de transición en las redes duales IPv4/IPv6. Es un mecanismo obsoleto.
- `::ffff:0:0/96` – La dirección IPv4 mapeada es usada como un mecanismo de transición en terminales duales.
- `fe80::/10` – El prefijo de enlace local, especifica que la dirección sólo es válida en el enlace físico local.
- `fec0::/10` – El prefijo de emplazamiento local, especifica que la dirección sólo es válida dentro de una organización local. LA RFC 3879 lo declaró obsoleto, estableciendo que los sistemas futuros no deben implementar ningún soporte para este tipo de dirección especial.
- `ff00::/8` – El prefijo de multicast es usado para las direcciones multicast.

Hay que resaltar que las direcciones de difusión (broadcast) no existen en IPv6, aunque la funcionalidad que prestan puede emularse utilizando la dirección multicast `FF01::1`, denominada “todos los nodos”.

1.2.4 DNS

A través de Internet, los ordenadores se conectan entre sí mediante sus respectivas direcciones IP. Sin embargo, a los seres humanos nos es más cómodo utilizar otra notación más fácil de recordar y utilizar, como los nombres de dominio; la traducción entre unos y otros se resuelve mediante los servidores de nombres de dominio DNS.

Un nombre de dominio o simplemente dominio representa un identificador común a un grupo de computadoras o equipos conectados a la red. Es una forma simple de dirección de Internet que está formado por un conjunto de caracteres (letras, números, guión). Es utilizado para localizar de una manera fácil los sitios en Internet ya que se puede asociar a la identidad de una persona, organización, empresa, idea, grupo, o a algún otro concepto.

Las siglas DNS son utilizadas para referirse al Domain Name System (Sistema de Nombres de Dominio) y/o al Domain Name Server (Servidor de Nombres de Dominio) de manera indistinta. Domain Name System es el sistema informático distribuido que establece para todo el Internet las correspondencias (resoluciones) entre nombres de dominio y direcciones IP. El Domain Name Server se refiere al equipo o los equipos de cómputo que forman parte del Domain Name System y que albergan la información y procesos para el establecimiento de las correspondencias (resoluciones) entre Nombres de Dominio y Direcciones IP para una o varias redes que se conectan a Internet.

El Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

La información del sistema de nombres de dominio está distribuida en diferentes servidores en el mundo. Existe una estructura jerárquica en la que cada servidor es responsable de conocer el nombre de dominio que administra y en caso de buscar una información que desconoce, debe dirigirse a la cabeza de la jerarquía para iniciar la búsqueda de la respuesta. Dicha cabeza son los llamados servidores raíz (root-servers) y son 13, la mayoría de ellos están en E.U. y se identifican por letras (de la A a la M), en México existe una copia del root-server F, el cual fue colocado por NIC México en coordinación con ISC (Internet Systems Consortium).

Componentes

Para la operación práctica del sistema DNS se utilizan tres componentes principales:

- 1) **Los Clientes DNS:** Un programa cliente DNS que se ejecuta en la computadora del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS (*Por ejemplo: ¿Qué dirección IP corresponde a nombre.dominio?*);
- 2) **Los Servidores DNS:** Que contestan las peticiones de los clientes. Los servidores recursivos tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada;
- 3) **Las Zonas de autoridad:** porciones del espacio de nombres de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad.

Partes de un nombre de dominio

Un nombre de dominio usualmente consiste en dos o más partes, técnicamente denominadas etiquetas, separadas por puntos cuando se escriben en forma de texto. Por ejemplo, `www.unam.mx` o `www.ingenieria.unam.mx`

A la etiqueta ubicada más a la derecha se le llama dominio de nivel superior (*Top Level Domain*).

Cada etiqueta a la izquierda especifica una subdivisión o subdominio. Subdominio expresa dependencia relativa, no dependencia absoluta. En teoría, esta subdivisión puede tener hasta 127 niveles, y cada etiqueta contener hasta 63 caracteres, pero restringido a que la longitud total del nombre del dominio no exceda los 255 caracteres, aunque en la práctica los dominios son casi siempre mucho más cortos.

Finalmente, la parte más a la izquierda del dominio suele expresar el nombre de la máquina (*hostname*). El resto del nombre de dominio simplemente especifica la manera de crear una ruta lógica a la información requerida.

Cada dominio o subdominio tiene una o más zonas de autoridad que publican la información acerca del dominio y los nombres de servicios de cualquier dominio incluido. La jerarquía de las zonas de autoridad coincide con la jerarquía de los dominios. Al inicio de esa jerarquía se encuentran los servidores raíz, es decir, los servidores que responden cuando se busca resolver un dominio de primer y segundo nivel.

La estructura de una base de datos DNS se asemeja mucho a un árbol invertido. Cada "hoja" o "nodo" de ese árbol es un dominio, comenzando todos los dominios desde el dominio root (el cual se denota con un punto ".").

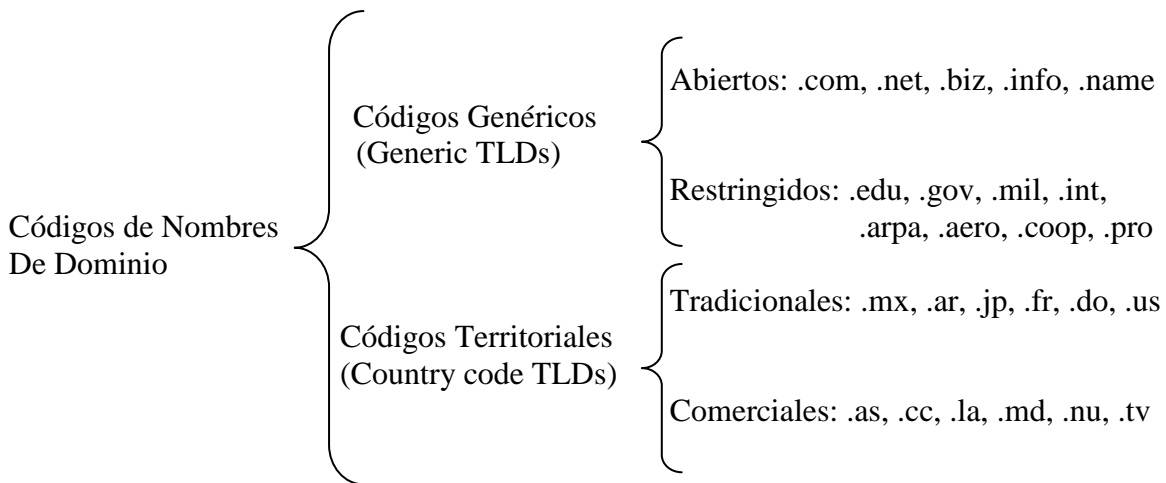


Figura 1.8 Clasificación de códigos de nombres de dominio.

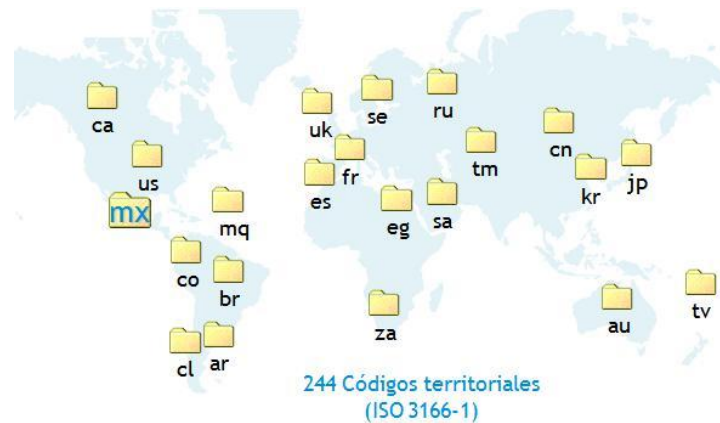


Figura 1.9 Códigos territoriales.

Tipos de resolución de nombres de dominio

Existen dos tipos de consultas que un cliente puede hacer a un servidor DNS:

1. **Iterativa.-** Las consultas iterativas, o resolución iterativa, consisten en la mejor respuesta que el servidor de nombres pueda dar. El servidor de nombres consulta sus datos locales (incluyendo su caché) buscando los datos solicitados.
2. **Recursiva.-** En las consultas recursivas el servidor repite el mismo proceso básico (consultar a un servidor remoto y seguir cualquier referencia) hasta que obtiene la respuesta a la pregunta.

Cuando existe más de un servidor autoritario para una zona, BIND utiliza el menor valor en la métrica RTT (*round-trip time*) para seleccionar el servidor. El RTT es una medida para determinar cuánto tarda un servidor en responder una consulta.

Tipos de registros DNS

- **A = Address** – (Dirección) Este registro se usa para traducir nombres de hosts a direcciones IP.
- **CNAME = Canonical Name** – (Nombre Canónico) Se usa para crear nombres de hosts adicionales, o alias, para los hosts de un dominio.
- **NS = Name Server** – (Servidor de Nombres) Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio. Cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres.
- **MX (registro) = Mail Exchanger** – (Registro de Intercambio de Correo) Asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio.
- **PTR = Pointer** – (Indicador) También conocido como 'registro inverso', funciona a la inversa del registro A, traduciendo IP's en nombres de dominio.
- **SOA = Start of authority** – (Autoridad de la zona) Proporciona información sobre la zona.

- **HINFO** = Host Information – (Información del sistema informático) Descripción del host, permite que la gente conozca el tipo de máquina y sistema operativo al que corresponde un dominio.
- **TXT** = Text - (Información textual) Permite a los dominios identificarse de modos arbitrarios.
- **LOC** = Localización - Permite indicar las coordenadas del dominio.
- **SRV** = SeRVicios - Permite indicar los servicios que ofrece el dominio. RFC 2782
- **SPF** = Sender Policy Framework - Ayuda a combatir el Spam. En este record se especifica cual o cuales hosts están autorizados a enviar correo desde el dominio dado.

Tipos de servidores DNS: Bind, PowerDNS, MaraDNS, Djbdns, Pdnss, MyDNS, DNS (Windows 2000/2003).

1.2.5 VOZ SOBRE IP

Voz sobre Protocolo de Internet, también llamado Voz sobre IP, VoIP, Telefonía IP, Telefonía por Internet, Telefonía Broadband y Voz sobre Broadband es el enrutamiento de conversaciones de voz sobre Internet o a través de alguna otra red basada en IP.

Los Protocolos que son usados para llevar las señales de voz sobre la red IP son comúnmente referidos como protocolos de Voz sobre IP o protocolos IP. Ellos pueden ser vistos como implementaciones comerciales de la Red experimental de Protocolo de Voz (1973) inventado por ARPANET.

Es una tecnología que permite la transmisión de la voz a través de redes IP en forma de paquetes de datos. La Telefonía IP es una aplicación inmediata de esta tecnología, de forma que permita la realización de llamadas telefónicas ordinarias sobre redes IP u otras redes de paquetes utilizando un PC, gateways y teléfonos estándares. En general, servicios de comunicación - voz, fax, aplicaciones de mensajes de voz - que son transportados vía redes IP, Internet normalmente, en lugar de ser transportados vía la red telefónica convencional.

En general, el servicio de telefonía vía VoIP es gratuito, o bien, su costo es mucho menor que el servicio equivalente tradicional y similar a la alternativa que los proveedores del servicio de la Red Pública Telefónica Conmutada (PSTN) ofrecen. Esto es debido a que se utiliza una misma red para llevar voz y datos, especialmente cuando los usuarios tienen sin utilizar toda la capacidad de una red ya existente la cual pueden usar para VoIP sin un costo adicional. Las llamadas de VoIP a VoIP entre cualquier proveedor son generalmente gratis, en contraste con las llamadas de VoIP a PSTN que generalmente cuestan al usuario de VoIP.

Hay dos tipos de servicio de PSTN a VoIP: Llamadas Locales Directas (Direct Inward Dialling: DID) y Números de acceso. DID conecta a quien hace la llamada directamente al usuario VoIP mientras que los Números de Acceso requieren que este introduzca el número de extensión del usuario de VoIP. Los Números de acceso son usualmente cobrados como

una llamada local para quien hizo la llamada desde la PSTN y gratis para el usuario de VoIP.

VoIP facilita gran número de tareas gracias a su funcionalidad.

El Estándar VoIP (H323)

Definido en 1996 por la UIT (Unión Internacional de Telecomunicaciones) proporciona a los diversos fabricantes una serie de normas con el fin de que puedan evolucionar en conjunto.

Por su estructura el estándar proporciona las siguientes ventajas:

- Permite el control del tráfico de la red, por lo que se disminuyen las posibilidades de que se produzcan caídas importantes en el rendimiento. Las redes soportadas en IP presenta las siguientes ventajas adicionales:
 - Es independiente del tipo de red física que lo soporta. Permite la integración con las grandes redes de IP actuales.
 - Es independiente del hardware utilizado.
 - Permite ser implementado tanto en software como en hardware, con la particularidad de que el hardware supondría eliminar el impacto inicial para el usuario común.
 - Permite la integración de Video y TPV

La industria de Voz sobre IP se encuentra en una etapa de crecimiento rápido. La evolución del uso de Voz sobre IP vendrá con la evolución de la infraestructura y de los protocolos de comunicación. En el año 2010, una cuarta parte de las llamadas mundiales se basarán en IP.

1.3 SISTEMAS OPERATIVOS

Un sistema operativo (SO) es un software de sistema, es decir, un conjunto de programas de computadora destinados a permitir una administración eficaz de sus recursos. Gestiona el hardware de la máquina desde los niveles más básicos, permitiendo la interacción con el usuario ya que actúa como intermediario entre éste y la computadora, proporcionando de esta forma un entorno en el cual el usuario pueda ejecutar programas.

1.3.1 FUNCIONES BÁSICAS DE UN SISTEMA OPERATIVO

Los sistemas operativos, en su condición de capa software que posibilitan y simplifican el manejo de la computadora, desempeñan una serie de funciones básicas esenciales para la gestión del equipo. Entre las más destacables, cada una ejercida por un componente interno (módulo en núcleos monolíticos y servidor en micronúcleos), se encuentran las siguientes:

- Proporcionar comodidad en el uso de un computador.
- Gestionar de manera eficiente los recursos del equipo, ejecutando servicios para los procesos (programas).

- Brindar una interfaz al usuario, ejecutando instrucciones (comandos).
- Permitir que los cambios debidos al desarrollo del propio SO se puedan realizar sin interferir con los servicios que ya se prestaban (evolutividad).

Un sistema operativo desempeña 5 funciones básicas en la operación de un sistema informático, las cuales se presentan a continuación:

1) Interfaces del usuario

Es la parte del sistema operativo que permite comunicarse con él de tal manera que se puedan cargar programas, acceder archivos y realizar otras tareas. Existen tres tipos básicos de interfaces: las que se basan en comandos, las que utilizan menús y las interfaces gráficas de usuario.

2) Administración de recursos

Sirven para administrar los recursos de hardware y de redes de un sistema informático, como el CPU, memoria, dispositivos de almacenamiento secundario y periféricos de entrada y de salida.

3) Administración de archivos

Un sistema de información contiene programas de administración de archivos que controlan la creación, borrado y acceso de archivos de datos y de programas. También implica mantener el registro de la ubicación física de los archivos en los discos magnéticos y en otros dispositivos de almacenamiento secundarios.

4) Administración de tareas

Los programas de administración de tareas de un sistema operativo administran la realización de las tareas informáticas de los usuarios finales. Los programas controlan que áreas tienen acceso al CPU y por cuánto tiempo. Las funciones de administración de tareas pueden distribuir una parte específica del tiempo del CPU para una tarea en particular, e interrumpir al CPU en cualquier momento para sustituirla con una tarea de mayor prioridad.

5) Servicio de soporte

Los servicios de soporte de cada sistema operativo dependerán de la implementación particular de éste con la que estemos trabajando. Entre las más conocidas se pueden destacar las implementaciones de Unix, desarrolladas por diferentes empresas de software, los sistemas operativos de Apple Inc., como Mac OS X para las computadoras de Apple, los sistemas operativos de Microsoft, y las implementaciones de software libre, como Linux o BSD producidas por empresas, universidades, administraciones públicas, organizaciones sin fines de lucro y/o comunidades de desarrollo.

Estos servicios de soporte suelen consistir en:

- Actualización de versiones.
- Mejoras de seguridad.
- Inclusión de alguna nueva utilidad como un nuevo entorno gráfico, un asistente para administrar alguna determinada función, etc.

- Controladores para manejar nuevos periféricos.
- Corrección de errores de software.

1.3.2 COMPONENTES DE UN SISTEMA OPERATIVO

- **Gestión de procesos**

Un proceso es simplemente, un programa en ejecución que necesita recursos para realizar su tarea: tiempo de CPU, memoria, archivos y dispositivos de E/S. El SO es el responsable de:

- Crear y destruir los procesos.
- Parar y reanudar los procesos.
- Ofrecer mecanismos para que se comuniquen y sincronicen.

El corazón de un sistema operativo es el núcleo, un programa de control que reacciona ante cualquier interrupción de eventos externos y que da servicio a los procesos, creándolos, terminándolos y respondiendo a cualquier petición de servicio por parte de los mismos.

- **Gestión de la memoria principal**

La memoria es una gran tabla de palabras o bytes que se referencian cada una mediante una dirección única. Este almacén de datos de rápido accesos es compartido por la CPU y los dispositivos de E/S, es volátil y pierde su contenido en los fallos del sistema. El SO es el responsable de:

- Conocer qué partes de la memoria están utilizadas y por quién.
- Decidir qué procesos se cargarán en memoria cuando haya espacio disponible.
- Asignar y reclamar espacio de memoria cuando sea necesario.

- **Gestión del almacenamiento secundario**

Un sistema de almacenamiento secundario es necesario, ya que la memoria principal (almacenamiento primario) es volátil y además muy pequeña para almacenar todos los programas y datos. También es necesario mantener los datos que no convenga mantener en la memoria principal. El SO se encarga de:

- Planificar los discos.
- Gestionar el espacio libre.
- Asignar el almacenamiento.

- **El sistema de E/S**

Consiste en un sistema de almacenamiento temporal (caché), una interfaz de manejadores de dispositivos y otra para dispositivos concretos. El sistema operativo debe gestionar el almacenamiento temporal de E/S y servir las interrupciones de los dispositivos de E/S.

- **Sistema de archivos**

Los archivos son colecciones de información relacionada, definidas por sus creadores. Éstos almacenan programas (en código fuente y objeto) y datos tales como imágenes, textos, información de bases de datos, etcétera. El SO es responsable de:

- Construir y eliminar archivos y directorios.
- Ofrecer funciones para manipular archivos y directorios.
- Establecer la correspondencia entre archivos y unidades de almacenamiento.
- Realizar copias de seguridad de archivos.

Existen diferentes Sistemas de Archivos, es decir, existen diferentes formas de organizar la información que se almacena en las memorias (normalmente discos) de los ordenadores. Por ejemplo, existen los sistemas de archivos FAT, FAT32, EXT2, NTFS.

- **Sistemas de protección**

Mecanismo que controla el acceso de los programas o los usuarios a los recursos del sistema. El SO se encarga de:

- Distinguir entre uso autorizado y no autorizado.
- Especificar los controles de seguridad a realizar.
- Forzar el uso de estos mecanismos de protección.

- **Sistema de comunicaciones**

Para mantener las comunicaciones con otros sistemas es necesario poder controlar el envío y recepción de información a través de las interfaces de red. También hay que crear y mantener puntos de comunicación que sirvan a las aplicaciones para enviar y recibir información, y crear y mantener conexiones virtuales entre aplicaciones que están ejecutándose localmente y otras que lo hacen remotamente.

- **Intérprete de órdenes**

El shell del sistema es el principal componente del SO que utiliza el usuario. Este uso se realiza siempre directa o indirectamente a través del intérprete. Generalmente incorpora un lenguaje de programación para automatizar las tareas.

Hay dos tipos de intérpretes de órdenes:

- 1) **Alfanuméricos:** las órdenes se expresan mediante un lenguaje específico usando las cadenas de caracteres introducidas por el terminal.
- 2) **Gráficos:** normalmente las órdenes se especifican por medio de iconos y otros elementos gráficos.

- **Programas de sistema**

Son aplicaciones de utilidad que se suministran con el SO pero no forman parte de él. Ofrecen un entorno útil para el desarrollo y ejecución de programas, siendo algunas de las tareas que realizan:

- Manipulación y modificación de archivos.
- Información del estado del sistema.
- Soporte a lenguajes de programación.
- Comunicaciones.

Gestor de recursos

Como gestor de recursos, el Sistema Operativo administra:

- La CPU (Unidad Central de Proceso).
- Los dispositivos de E/S (entrada y salida)
- La memoria principal (o de acceso directo).
- Los discos (o memoria secundaria).
- Los procesos (o programas en ejecución) y en general todos los recursos del sistema.

1.3.3 CLASIFICACIÓN DE LOS SISTEMAS OPERATIVOS

Los sistemas operativos se clasifican en tres:

- 1) Por su estructura (visión interna)
- 2) Por los servicios que ofrecen y
- 3) Por la forma en que ofrecen sus servicios (visión externa).

1) Sistemas operativos por su estructura

❖ Estructura monolítica

Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra. Las características fundamentales son:

- Construcción del programa final a base de módulos compilados separadamente que se unen a través del ligador.
- Buena definición de parámetros de enlace entre las distintas rutinas existentes, que puede provocar mucho acoplamiento.
- Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.
- Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

❖ *Estructura jerárquica*

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del software, del sistema operativo, donde una parte del sistema contenía subpartes y esto organizado en forma de niveles.

Se constituyó una estructura jerárquica o de niveles en los sistemas operativos, por lo que también son denominados “sistemas multicapa”. Ejemplos: Multics y Unix.

❖ *Máquina virtual*

Se trata de un tipo de sistemas operativos que presentan una interface a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la multiprogramación y la máquina extendida. El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.

❖ *Cliente-servidor (Microkernel)*

El tipo más reciente de sistemas operativos es el denominado Cliente-Servidor, que puede ser ejecutado en la mayoría de las computadoras, ya sean grandes o pequeñas.

Este sistema sirve para toda clase de aplicaciones por tanto, es de propósito general y cumple con las mismas actividades que los sistemas operativos convencionales.

El núcleo tiene como misión establecer la comunicación entre los clientes y los servidores. Los procesos pueden ser tanto servidores como clientes.

2) **Sistemas operativos por servicios**

Esta clasificación es la más comúnmente usada y conocida desde el punto de vista del usuario final.

- **Por número de usuarios**

- a. ***Sistema operativo monousuario***

- Son aquéllos que soportan a un usuario a la vez no importando la cantidad de procesadores o procesos que el usuario esté ejecutando en un mismo instante de tiempo. Las computadoras personales se han clasificado en esta sección.

- b. ***Sistema operativo multiusuario***

- Son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

- **Por el número de tareas**
 - a. ***Sistema operativo monotarea***
Son aquellos sistemas que solo permiten realizar una sola tarea a la vez por usuario.
 - b. ***Sistema operativo multitarea***
Son sistemas que permiten realizar varias actividades o procesos al mismo tiempo sin ningún problema. Este tipo de SO normalmente asigna los recursos disponibles (CPU, memoria, periféricos) de forma alternada a los procesos que los solicitan, de manera que el usuario percibe que todos funcionan a la vez, de forma concurrente.

- **Por el número de procesadores**
 - a. ***Sistema operativo de uniprocreso***
Es aquel sistema que solo puede trabajar con un solo procesador, si tuviera otro sería obsoleto. Ejemplos: Dos y MacOs.
 - b. ***Sistema operativo de multiprocreso***
Este sistema puede utilizar varios procesadores para distribuir el trabajo de cada uno. Generalmente estos trabajan o pueden ser de dos tipos:
 - ***Asimétrica:*** es donde el sistema operativo selecciona un procesador maestro y los demás funcionan como esclavos.
 - ***Simétrica:*** en este tipo se envía información o se trabaja con el procesador con menos carga y así se distribuye mejor el trabajo.

3) Sistemas operativos por la forma de ofrecer sus servicios

Esta clasificación también se refiere a una visión externa, que en este caso se refiere a la del usuario, el cómo accesa a los servicios. Bajo esta clasificación se pueden detectar dos tipos principales: sistemas operativos de red y sistemas operativos distribuidos.

❖ ***Sistema operativo de red***

Los sistemas operativos de red se definen como aquellos que tiene la capacidad de interactuar con sistemas operativos en otras computadoras por medio de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades. El usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que desee acceder.

❖ ***Sistema operativo distribuido***

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso) en una sola máquina virtual que el usuario accesa en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por

nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo habitual.

1.3.4 UNIX

En los años 60, un proyecto estaba bajo la línea del Instituto Tecnológico de Massachusetts (MIT), para mejorar el estado de habilidad con software de tiempo compartido (timesharing). En este proyecto fueron colaboradores el MIT, los laboratorios Bell, y General Electric.

En 1966, un grupo de investigadores de los Laboratorios Bell desarrolló un sistema operativo experimental llamado MULTICS (Información multiplexada y Sistema de Computación). Fue diseñado como sistema operativo interactivo para una computadora de la General Electric (GE 645), permitiendo compartir la información y a la vez daba la seguridad que los investigadores buscaban. Sin embargo hubo problemas en el desarrollo que hicieron muy lentas las versiones y con mucho requerimiento de memoria, debido a esto el proyecto fue abandonado.

En 1969, Ken Thompson y Dennis Ritchie decidieron diseñar un sistema operativo que llenara los requisitos predispuestos. Ellos crearon un sistema operativo multitarea, con sistema de archivos, intérprete de órdenes y algunas utilidades para el PDP-7 de DEC. Al sistema se le llamó con el nombre de UNICS (Información Uniplexada y Sistema de Computación), éste tenía la característica de tiempo compartido (timesharing), pues podía soportar dos usuarios simultáneamente. Ese nombre y un juego de palabras sobre MULTICS, hicieron que en 1970, Brian Kernighan le diera el nombre de UNIX.

En 1972 había varias computadoras con Unix y se esperaban más. Algún tiempo después Ritchie desarrolló el lenguaje de programación C. En 1973, Ritchie y Thompson escribieron el núcleo de Unix en C, un lenguaje de programación de alto nivel, a diferencia de la mayoría de los sistemas escritos generalmente en ensamblador. En un comienzo Thompson pensó escribir Unix en Fortran, pero encontró dificultades con ese lenguaje debido a su falta de portabilidad. Gracias a lo portable, innovador y estar escrito en un lenguaje de alto nivel modificable de acuerdo a preferencias personales, Unix ha alcanzado el nivel de popularidad que tiene hoy en día. Además, con este nuevo enfoque fue posible portar el sistema operativo a otras máquinas sin muchos cambios, solamente efectuando una nueva compilación en la máquina destino.

Inicialmente UNIX fue considerado como un proyecto de investigación, hasta el punto de distribuirse de forma gratuita en algunas universidades, pero después la demanda del producto hizo que los laboratorios Bell iniciaran su distribución oficial.

Características

- Multiusuario
- Multitarea
- Multiprocesador

- De propósito general
- Portable
- Independencia de dispositivos
- Intérprete de órdenes programable e intercambiable
- Mecanismos de comunicación:
 - Entre usuarios
 - Entre procesos
- Facilita compartir datos y programas
- Seguro
- Interactivo
- Dispone de un lenguaje de control programable llamado SHELL
- Ofrece facilidades para la creación de programas y sistemas y el ambiente adecuado para las tareas de diseños de software.
- Emplea manejo dinámico de memoria por intercambio o paginación.
- Tiene capacidad de interconexión de procesos y permite comunicación entre éstos.
- Emplea un sistema jerárquico de archivos, con facilidades de protección de archivos, cuentas y procesos.
- Tiene facilidad para redireccionamiento de Entradas/Salidas.

Las dos variantes fundamentales: System V y BSD

La Universidad de California en Berkeley comenzó sus desarrollos en el campo UNIX, añadiendo nuevas características y haciendo modificaciones. Así, en 1975 Ken Thompson promovió el desarrollo y sacó a la luz su propia versión de UNIX, conocida como BSD. Desde entonces BSD pasó a convertirse en la gran competidora de los laboratorios Bell. En esta versión contribuyeron Bill Joy y Chuck Haley, sus contribuciones son numerosas pero entre ellas destacan un compilador de Pascal, el editor vi, el editor ex y el Shell C. Más tarde, a principios de los 80, se produjo el lanzamiento de Sistema III, la primera versión comercial del sistema operativo UNIX. En 1983 AT&T introdujo el UNIX Sistema V versión 1. En ese mismo año, Berkeley lanza una poderosa versión UNIX conocida como BSD versión 4.2. Entre sus características principales se encuentran la gestión de archivos muy sofisticada así como la posibilidad de trabajo en redes basadas en los protocolos TCP/IP. Esta versión de UNIX la adoptaron varios fabricantes, entre ellos Sun Microsystems, lo que dio lugar al conocido sistema SunOS.

Familias UNIX

- **AT&T:** la familia que tuvo su origen en el UNIX de AT&T. Considerada la familia UNIX "pura" y original. Sus sistemas operativos más significativos son UNIX System III y UNIX System V.
- **BSD:** familia originada por el licenciamiento de UNIX a Berkeley. BSD incorpora propiedad intelectual no originaria de AT&T, la primera implementación de los protocolos TCP/IP que dieron origen a Internet.
- **AIX:** esta familia surge por el licenciamiento de UNIX System III a IBM.
- **Xenix:** familia derivada de la adquisición de los derechos originales de AT&T primero por parte de Microsoft y de esta los vendió a SCO.

- **GNU:** En 1983, Richard Stallman anunció el Proyecto GNU, un ambicioso esfuerzo para crear un sistema similar a Unix, que pudiese ser distribuido libremente. El software desarrollado por este proyecto -por ejemplo, GNU Emacs y GCC - también han sido parte fundamental de otros sistemas UNIX.
- **Linux:** En 1991, cuando Linus Torvalds empezó a proponer el núcleo Linux y a reunir colaboradores, las herramientas GNU eran la elección perfecta. Al combinarse ambos elementos, conformaron la base del sistema operativo (basado en POSIX) que hoy conocemos como Linux. Las distribuciones basadas en el núcleo, el software GNU y otros agregados entre las que podemos mencionar a Red Hat Linux y Debian GNU/Linux se han hecho populares tanto entre los aficionados a la computación como en el mundo empresarial. Obsérvese que Linux tiene un origen independiente, por lo que se considera un 'clónico' de UNIX y no un UNIX en el sentido histórico.

A lo largo de la historia ha surgido una gran multitud de implementaciones comerciales de UNIX. Sin embargo, un conjunto reducido de productos han consolidado el mercado y prevalecen gracias a un continuo esfuerzo de desarrollo por parte de sus fabricantes. Los más importantes son:

- **Solaris de Sun Microsystems.** Uno de los sistemas operativos Unix más difundido en el entorno empresarial y conocido por su gran estabilidad. Parte del código fuente de Solaris se ha liberado con licencia de fuentes abiertas (OpenSolaris).
- **AIX de IBM.** El UNIX "propietario" de IBM ha cumplido 20 años de vida en el 2006 y continúa en pleno desarrollo, con una perceptible herencia del mainframe en campos como la virtualización o la RAS de los servidores, heredada de sus "hermanos mayores".
- **HP-UX de Hewlett-Packard.** Este sistema operativo también nació ligado a las computadoras departamentales de este fabricante. También es un sistema operativo estable que continua en desarrollo.

Adicionalmente, existen distribuciones de Linux muy difundidas en la empresa. Se trata de distribuciones sostenidas por grandes compañías tales como:

- **SuSE Linux de Novell.** Originalmente liberado por la compañía alemana SuSE. Es popular por sus herramientas de administración centralizada.

Las siguientes implementaciones de UNIX tienen importancia desde el punto de vista histórico, no obstante, actualmente están en desuso:

- **Tru64 UNIX** actualmente de Hewlett-Packard (antes de Compaq y originalmente de Digital Equipment Corporation).
- **UnixWare y SCO OpenServer** anteriormente de Santa Cruz Operation y ahora de SCO Group.
- **UX/4800** de NEC.
- **IRIX** de Silicon Graphics Inc.

1.3.5 LINUX

Linux es un sistema operativo tipo Unix (también conocido como GNU/Linux) que se distribuye bajo la Licencia Pública General de GNU (GNU GPL), es decir que es software libre. Su nombre proviene del Núcleo de Linux, desarrollado en 1991 por Linus Torvalds.

Linux nació gracias a la idea de Linus Torvalds de crear un sistema basado en Unix para máquinas i386. La historia de Linux está fuertemente vinculada a la del proyecto GNU. El proyecto GNU, iniciado en 1983, tiene como objetivo el desarrollo de un sistema Unix completo compuesto enteramente de software libre. Hacia 1991, cuando la primera versión del núcleo Linux fue liberada, el proyecto GNU había producido varios de los componentes del sistema operativo, incluyendo un intérprete de comandos, una biblioteca C y un compilador, pero aún no contaba con el núcleo que permitiera completar el sistema operativo.

El núcleo fue creado por Linus Torvalds a principios de los 90. Cuando tuvo una versión suficientemente estable comenzó a distribuirla bajo la licencia GPL y solicitó ayuda para hacer pruebas y mejorarlo.

El éxito de Linux se debe fundamentalmente a su distribución por Internet ya que miles de programadores voluntarios alrededor del mundo han participado en el proyecto, mejorándolo continuamente. Torvalds y otros desarrolladores de los primeros días de Linux adaptaron los componentes de GNU y de BSD, así como de otros muchos proyectos como Perl, Apache, Python, etc. para trabajar con el núcleo Linux, creando un sistema operativo completamente funcional procedente de muchísimas fuentes diferentes, la mayoría libres.

Linux es usado ampliamente en servidores y supercomputadores, y cuenta con el respaldo de corporaciones como Dell, Hewlett-Packard, IBM, Novell, Oracle, Red Hat y Sun Microsystems. Puede ser instalado en gran variedad de hardware, incluyendo computadores de escritorio y portátiles (PCs x86 y x86-64 así como Macintosh y PowerPC), computadores de bolsillo, teléfonos celulares, dispositivos empotrados, videoconsolas (Xbox, PlayStation 3, PlayStation Portable, Dreamcast, GP2X) y otros (como máquinas de juegos, enrutadores y algunos modelos de dispositivos iPod).

Las variantes de estos sistemas se denominan "distribuciones" y su objetivo es ofrecer una edición que cumpla con las necesidades de determinado grupo de usuarios. Algunas son gratuitas y otras de pago, algunas insertan software no libre y otras contienen solo software libre.

Actualmente Linux cuenta con los principales gestores de ventanas, utilidades para internet, compiladores, editores, entre otras cosas. Linux se utiliza con éxito como servidor en muchas empresas y universidades de todo el mundo, y cada vez son más los usuarios particulares que se deciden por este sistema operativo.

En el siguiente cuadro se presentan las distribuciones de Linux más utilizadas:

Distribución	Descripción
DEBIAN	Es una distribución orientada más a desarrolladores y programadores. El énfasis de esta distribución es incluir en su sistema solamente software libre según la definición de la Fundación del Software Libre (FSF).
UBUNTU	Distribución basada en Debian, con lo que esto conlleva y centrada en el usuario final y facilidad de uso. Muy popular y con mucho soporte en la comunidad.
MANDRIVA (MANDRAKE)	Es una distribución originalmente basada en RedHat que se enfoca principalmente hacia la facilidad de uso.
REDHAT	Es la distribución más popular de Linux y para la que hay más paquetes comerciales de software. Está orientada tanto al desktop como a servidores. La mayoría de servidores web que utilizan Linux como sistema operativo usan esta distribución.
S.U.S.E	Es la distribución más popular en Europa y probablemente la segunda a nivel mundial. Está orientada tanto a desktops como a servidores.
SLACKWARE	Esta distribución pretende parecerse a BSD desde el punto de vista del administrador de sistemas. No es muy popular.
STAMPEDE	Es una distribución enfocada al rendimiento y velocidad del sistema.
CALDERA	El énfasis de esta distribución es la facilidad de uso e instalación para los usuarios. Se orientan más hacia el desktop, como cualquier otra distribución de Linux, puede ser usada para servidores.
COREL	Es una distribución basada en Debian, pero extensivamente modificada para hacerla tan fácil de usar como un conocido sistema operativo de Microsoft. Es quizá la distribución más fácil de utilizar para alguien que no esté familiarizado con el sistema.
FEDORA	Distribución para propósitos generales basada en Red Hat.
KNOPPIX	Distribución basada en Debian.

Tabla 1.13 Distribuciones de LINUX.

1.4 LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación son herramientas que nos permiten crear programas y software. La evolución de los lenguajes de programación se puede dividir en 5 generaciones, que se presentan a continuación:

Generación	Nombre	Particularidad
Primera	De máquina	Específico para cada microprocesador, uso de código binario
Segunda	Ensamblador	Uso de nemotécnicos que abstraen del lenguaje máquina.
Tercera	De procedimientos	Lenguajes estructurados con comandos cercanos al lenguaje común.
Cuarta	Orientados a procesos	Programas orientados a problemas específicos.
Quinta	Natural	Incluye inteligencia artificial y sistemas expertos.

Tabla 1.14 Las generaciones de lenguajes de programación.

1.4.1 LENGUAJES DE MARCAS

El lenguaje de marcado (markup language), también denominado lenguaje de anotaciones o de etiquetas o simplemente de marcas, se define como un conjunto de reglas para estructurar y dar formato a un documento electrónico. Suelen utilizar etiquetas para definir el inicio y el final de un elemento: un párrafo, un título, un elemento subrayado, etc.

❖ SGML

SGML (*Standard Generalized Markup Language*), o bien, lenguaje estándar generalizado de marcado, es un estándar internacional que permite definir lenguajes para dar formato a documentos. Este lenguaje es muy estable debido a que es flexible y rico en posibilidades, fue diseñado para permitir el intercambio de información entre distintas plataformas, soportes físicos, lógicos y diferentes sistemas de almacenamiento y presentación, independientemente de su grado de complejidad.

Funcionamiento de SGML

La función del etiquetado es aportar información que refleje la estructura jerárquica de un documento, de forma que ayude al lector u ordenador a procesar su contenido. Existen dos tipos de etiquetado:

1. **Etiquetado procedimental.**- técnica mediante la cual un operario utiliza instrucciones crípticas y dependientes del funcionamiento de un sistema determinado para que ejecute una acción, como por ejemplo activar la tipografía, poner en negrita, centrar, etc.
2. **Etiquetado descriptivo.**- que identifica los elementos estructurales de un documento, determinando su estructura lógica.

El lenguaje de etiquetado SGML permite distinguir entre el contenido (datos) y el etiquetado.

SGML utiliza un conjunto de caracteres basado en el estándar ASCII (*American Standard Coding for the Interchange of Information*).

En SGML todo el etiquetado es lógico, es decir, en lugar de utilizar códigos crípticos, se utilizan nombres de elementos, delimitados por marcas que indican el comienzo y final de los objetos lógicos.

Las definiciones de los elementos, denominadas declaraciones de elementos, tienen dos funciones:

1. Aportar el nombre de las etiquetas.
2. Describir el modelo de contenido (lo que cada elemento puede contener).

Componentes de un documento SGML

Un documento SGML actualmente está compuesto de tres archivos que trabajan de manera conjunta:

1. *La Declaración SGML*

La declaración SGML dice al usuario qué puede y qué no puede estar contenido en el documento SGML, qué caracteres serán usados, qué características específicas de SGML serán implementadas, y qué sintaxis se utilizará en el documento. Se trata de un diagrama formal y normalizado que le indica al sistema receptor el conjunto de caracteres, los delimitadores y las características opcionales de SGML que se están utilizando. La declaración SGML y la DTD trabajan de forma conjunta. Si la declaración SGML dice que una función en particular no podrá ser usada, la DTD debe respetarlo, de otra manera existirán errores en el documento.

2. *DTD (Declaración de Tipo de Documento)*

En la DTD (Document Type Definition) o Definición del Tipo de Documento se identifica la estructura del documento, es decir, aquellos elementos que son necesarios en la elaboración de un documento o un grupo de documentos estructurados de manera similar. Contiene las reglas de dichos elementos: el nombre, su significado, dónde pueden ser utilizados y qué pueden contener. Si la DTD se almacena en un archivo separado (con la extensión .dtd), éste se puede referenciar sin residir dentro del documento SGML, porque es corriente que se sustituya la DTD completa por una línea que indique que la DTD se edita como un texto público o se encuentra ya disponible en el sistema receptor. Sin la DTD (o una referencia a él), el documento SGML no será validado apropiadamente por el parser. El parser es una herramienta que asegura la adecuada conformación de SGML dentro de las múltiples especificaciones que un documento SGML puede tener. Un documento no será validado por el parser si no cumple con las especificaciones de la DTD, por ejemplo, si carece de título, ya que éste es un elemento obligatorio.

En la definición del tipo de documento o DTD (Document Type Definition) se utilizan los comandos:

- **ELEMENT**: sirve para definir una etiqueta.
- **ENTITY**: con este comando se pueden indicar ciertos elementos del texto en forma de sucesiones de caracteres ASCII y también utilizar dentro del texto los caracteres reservados para la sintaxis de SGML, como los signos 'menor que' o 'mayor que'.
- **ATTLIST**: permite establecer atributos complementarios para determinados elementos concretos. Todos los atributos pertenecientes a un elemento se resumen aquí en una sola lista en la que se establecen los nombres de los atributos, así como los valores autorizados de cada atributo.

Otro concepto importante dentro del SGML y del DTD es el de secciones marcadas. Éstas son usadas para aislar ciertos trozos de texto como específicos de una arquitectura, y se inician con los caracteres: "". Las claves para la especificación de las secciones marcadas son INCLUDE, IGNORE, CDATA, RCDATA o TEMP.

Otros comandos importantes son:

- USEMAP y SHORTREF: sirven para establecer abreviaturas
- NOTATION: para fijar anotaciones en la hoja de estilo.
- DOCTYPE: sirve para incluir un DTD o para llamar a un archivo con la extensión .dtd que lo contiene.

Para escribir la DTD hay que analizar los documentos y tener en cuenta los posibles cambios (revisiones, borrado, entre otros). Además, las etiquetas definidas en la DTD servirán para que editores, motores de búsqueda y visualizadores interpreten los documentos SGML.

3. *Instancia o muestra de documento*

La Muestra de Documento contiene el marcado o etiquetado y el contenido del texto, es decir, contiene el documento en sí que incluye tanto el texto como el marcado.

El contenido del documento tiene objetos SGML que siguen la estructura del árbol definido en la DTD. Los objetos SGML son fundamentalmente dos:

- Elementos (insertados como etiquetas y sus atributos locales)
- Entidades (para caracteres especiales, texto almacenado y archivos externos).

El siguiente ejemplo es una pequeña muestra de cómo se hace el marcado de texto:

```
<titulo>CONCEPTOS GENERALES DE SGML</titulo>
<autor> CORAL MONTES </autor>
<contenido>
<capitulo>
<subtitulo><DEFINICION DE SGML</subtitulo>
<parrafo> SGML es un acrónimo de Standard Generalized Markup
Lenguaje o Lenguaje de Señalización General Normalizado.
</parrafo>
</capitulo>
</contenido>
```

Las etiquetas están delimitadas por los caracteres < > para abrir y por </ > para cerrar.

❖ **HMTL**

HTML, acrónimo inglés de HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto es un lenguaje que posibilita la creación y edición de documentos web de manera sencilla ya que permite describir hipertexto, es decir, texto presentado de forma estructurada y formato definido a través de etiquetas, las cuales dan lugar a enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas e inserciones multimedia (gráficos, sonido, etc.). Básicamente, permite definir la forma en que un documento será visualizado en la web.

Para la creación de documentos HTML es necesario:

- a) **Un procesador de texto:** Los documentos HTML están en formato de texto sencillo o texto plano (también conocido como código ASCII). El procesador de texto se utiliza para escribir el documento en lenguaje HTML, que será posteriormente interpretado por el programa navegador correspondiente, siempre que el documento esté guardado en formato: "sólo texto".

El texto escrito está compuesto de dos partes: el contenido de la información y el conjunto de etiquetas del lenguaje HTML, que permiten determinar el estilo y el tipo de letra que tendrá la presentación del documento final y que pueda ser leído por un programa cliente.

- b) **Un navegador del WWW** o lo que se denomina programa cliente que permite el acceso a páginas WWW de Internet.

Por convención, archivos de formato HTML usan la extensión .htm o .html.

Las etiquetas del lenguaje HTML

HTML es un metalenguaje heredado del SGML, y por lo tanto usa etiquetas o directivas (*tags*) para indicarle al programa cliente de Web cómo mostrar el texto. Además, una de sus ventajas es que puede ser implementado por código de otros lenguajes como JavaScript o Visual Basic Script que amplían y mejoran su capacidad original.

Las etiquetas están formadas por determinados caracteres ubicados entre los signos < >, y con la barra </> cuando se trata de la segunda etiqueta de un efecto (la de cierre)

Estructura de un documento HTML

Todas las páginas web tienen la siguiente estructura:

```
<html>
<head>
<title>Mi primera página</title>
</head>
<body>
</body>
</html>
```

Donde las etiquetas utilizadas se refieren a lo siguiente:

- **<html>**: Es la etiqueta que define el inicio del documento html, le indica al navegador que todo lo que viene a continuación debe tratarlo como una serie de códigos html.
- **<head>**: Define la cabecera del documento html, esta cabecera suele contener información sobre el documento que no se muestra directamente en el navegador.

- **<body>**: Define el contenido principal o cuerpo del documento, ésta es la parte del documento html que se muestra en el navegador, dentro de esta etiqueta pueden definirse propiedades comunes a toda la página.
- **<title>**: Dentro del encabezado hay información que no se ve en la pantalla principal pero en éste puede ir el título documento. El título debe ser breve y descriptivo de su contenido, ya que es lo que veremos como título de la ventana en los navegadores que lo permitan y como se conocerá nuestra página en algunos buscadores y en la lista de favoritos de los usuarios cuando añadan nuestra página a su bookmark (o agenda de direcciones).

Cabe resaltar que dentro del cuerpo del documento se pueden utilizar una serie de etiquetas para definir distintos elementos, tales como: el tamaño o tipo de letra, el color de fondo y márgenes, los caracteres especiales, entre otros, Asimismo, los enlaces, listas, tablas e imágenes, etc.

❖ XML

XML (Extensible Markup Language) o Lenguaje de Marcas Generalizado es un lenguaje para definir lenguajes. Es más sencillo y más fácil de aplicar que SGML, y fue diseñado para hacer frente al gran y rápido desarrollo de las nuevas tecnologías de Internet.

En XML no hay elementos. Cada usuario puede crear su propio lenguaje para el formato de datos y documentos según sus necesidades, siguiendo las reglas de XML a partir de unos estándares generales.

XML define etiquetas en función del tipo de datos que está describiendo y no de la apariencia final que tendrán en pantalla o en la copia impresa, además de permitir definir nuevas etiquetas y ampliar las existentes. XML describe el contenido de lo que etiqueta. Su objetivo es conseguir una página Web más semántica.

Documentos XML

Las herramientas de creación XML se utilizan para crear y editar documentos XML, la más simple de todas es el editor de textos aunque es recomendable utilizar un editor específico. Dentro de éstos, existen 2 tipos principales:

1. Los que representan los archivos en forma de árbol y nos permiten construir nuestro documento trabajando sobre este árbol y formularios adicionales. Ejemplos: SoftQuadXMetal, XML Notepad de Microsoft, EditML, Visual XML, XMLwriter.
2. Los que representan el documento XML en su formato original y que normalmente son editores de archivos de texto con facilidades para XML. Ejemplos: XED y PSGML de Emacs.

Entre ambos tipos hay que diferenciar los que trabajan con una DTD y por lo tanto validan el contenido de lo que escribimos y los que simplemente nos aseguran que el documento

XML esté bien formado, es decir, sintácticamente correcto respecto de las especificaciones de éste.

Requisitos para que un documento sea considerado válido

- El documento debe estar bien construido.
- El elemento root (raíz) debe coincidir con el nombre que proporcione la declaración de tipo de documento (DTD).
- El documento debe tener una DTD que declara todos los elementos, atributos y entidades que se utilicen en el documento.
- El documento debe adherirse a la gramática que establezca la DTD.

Requisitos principales para que un documento se considere bien construido

- Debe tener un elemento de documento en el que se encuentren todos los demás elementos, o sea un único elemento raíz.
- Todo elemento deberá tener una etiqueta de inicio y su correspondiente etiqueta de cierre, o una sola etiqueta vacía.
- Todos los elementos deberán estar correctamente anidados.
- Todos los valores de atributos deberán ir entre comillas.
- Todas las entidades que se utilicen deberán declararse en la DTD.
- XML distingue entre mayúsculas y minúsculas. Puede utilizar los dos tipos de letra pero no los considera iguales.

Componentes de marcado XML

- Etiquetas de elementos.
- Instrucciones de procesamiento.
- Declaraciones de tipos de documentos.
- Referencias de entidades.
- Comentarios.
- Secciones marcadas.

Estructura del XML

En un documento XML se distinguen 2 estructuras: física y lógica.

La estructura física está dada por una o más unidades de almacenamiento virtual, llamadas entidades. Todas estas unidades tienen contenido y todas ellas (excepto la entidad documento y el subconjunto externo del DTD) están identificadas por un nombre. Cada documento XML contiene una entidad, llamada raíz del documento, que sirve como punto de partida para el procesador XML y puede contener el documento completo.

La estructura lógica está determinada por los elementos que hay en ese documento junto con las relaciones que hay entre tales elementos, los cuales están delimitados por etiquetas

de comienzo y de final o, en el caso de elementos vacíos, por una etiqueta de elemento vacío.

La estructura de un archivo XML es muy similar a la estructura en HTML sólo que con la gran diferencia de que nosotros nombramos las etiquetas para así poder almacenar información que nos interesa para después lograr un acceso a esta. Ejemplo:

```
<?xml versión="1.0" encoding="UTF-8" standalone="yes"?>
<ficha>
<nombre>Coral Alejandra</nombre>
<apellidos>Montes Carmona</apellidos>
<direccion>Azul #23 Colonia 7 mares</direccion>
</ficha>
```

Son varios los vocabularios desarrollados en XML con el fin de ampliar sus aplicaciones. Podemos considerar fundamentales: XHTML, XSL-FO y XSLT, XLink, XPointer y Schema. Además, existen también versiones para usos específicos, como MathML (fórmulas matemáticas), SVG (gráficos vectoriales), DocBook (lenguaje para el formato de libros electrónicos), RSS (sindicación de noticias), o XBRL (partes financieros).

1.4.2 JAVA

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las librerías de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Tecnología JAVA

Como un ambiente de desarrollo, la tecnología Java provee un conjunto de herramientas: un compilador, un intérprete, un generador de documentación y un conjunto de paquetes que contienen las clases que nos sirven para crear todo tipo de aplicaciones.

Versiones de JAVA

Esta tecnología está formada por tres versiones:

- 1) **JAVA SE**, Java Standard Edition.- conocida también como J2SDK, Standard Development kit. Para desarrollo y deployment de aplicaciones Java, es el fundamento para J2EE.
- 2) **JAVA EE**, Java Enterprise Edition.- Para desarrollo y deployment en un servidor de aplicaciones empresariales basadas en Servlets, Java Server Pages y Enterprise Java Beans.
- 3) **JAVA ME**, Java Micro Edition.- Para desarrollo y deployment de aplicaciones de Java en dispositivos móviles, como teléfonos, PDAs, TVs, etc.

Programas JAVA

A continuación se listan algunos de los tipos de programas que podemos generar por medio de la tecnología Java:

- **Aplicaciones.-** Programas convencionales que corren bajo control del sistema operativo.
- **Applets.-** Programas que corren bajo un browser de web (Explorer o Netscape).
- **Java Beans.-** Componentes (en muchos casos gráficos) que siguen una serie de convenciones preestablecidas.
- **Servlets.-** Aplicaciones que se ejecutan en un servidor de aplicaciones y manejan, en general, la presentación gráfica del sistema.
- **JSPs (Java Server Pages).-** Elementos tipo HTML que son convertidos a Servlets por el servidor de aplicaciones.
- **EJBs (Enterprise Java Beans).-** Aplicaciones que se ejecutan en un servidor de aplicaciones que implementan generalmente la lógica empresarial del sistema.

Objetivos primarios de Java

- **Facilidad de uso.-** ya que tiene la sintaxis basada en C++, la semántica basada en Smalltalk, pero simplificando ambos lenguajes, además de que es autodocumentable y corre en cualquier ambiente que soporte la máquina virtual de java.
- **Portable entre diversas plataformas.-** Esto es logrado por medio del compilador e intérprete de esta tecnología.
- **Orientado a objetos.-** Porque permite representar situaciones de la vida real naturalmente, además de que en Java todo se basa en clases y objetos.
- **Multithreading.-** Porque de esta manera permite aprovechar la velocidad del procesador, así como crear programas complejos y profesionales fácilmente.

Java cumple estos objetivos debido a tres mecanismos principales:

1) Máquina Virtual de Java (JVM)

Una definición es la siguiente: “Máquina imaginaria que es implementada ya sea mediante emulación de software o en una máquina real. El código que ejecuta la máquina virtual se encuentra en archivos .class que son resultado de la compilación del programa fuente .java”, sus instrucciones son los “byte codes” que son resultado de esta compilación, las implementaciones de esta máquina virtual en diferentes sistemas operativos es lo que hace que la tecnología Java sea multiplataforma (Sun Microsystems SL-275, 2006).

2) Recolección de basura

Algunos lenguajes de programación permiten la asignación de memoria dinámica en tiempo de ejecución. El proceso de asignación de memoria varía basado en la sintaxis del lenguaje pero siempre regresa un apuntador a la dirección de inicio del bloque de memoria reservado. En algunos lenguajes la liberación de la memoria es responsabilidad del programador, esto puede tener un grado de dificultad enorme porque no siempre se conoce cuando la memoria debe de ser liberada, a consecuencia de ello los programas pueden tener pérdida de memoria. El lenguaje de programación Java le quita la responsabilidad al programador de liberar la memoria, ya que la JVM, provee un thread de bajo nivel llamado recolector de basura (Garbage Collection), el cual se encarga durante los ciclos inactivos de la JVM de buscar alguna memoria que pueda ser liberada, para liberarla (Sun Microsystems SL-275, 2006).

3) Seguridad en el código

Cuando se ejecuta un programa, JVM puede monitorear estrictamente lo que va ocurriendo, desde el momento de cargar una clase que necesite el programa en ejecución, así como que no viole la integridad del sistema, por lo tanto la tecnología java es muy segura (Sun Microsystems SL-275, 2006).

1.4.3 JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación. Utilizado principalmente en páginas web embebido en el código HTML, con una sintaxis semejante a la del lenguaje Java y el lenguaje C, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, la que desarrolló los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar operaciones y en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Inicialmente los autores lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995. En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la European Computer Manufacturers Association ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también como un estándar ISO.

JScript es la implementación de ECMAScript de Microsoft, muy similar al JavaScript de Netscape, pero con ciertas diferencias en el modelo de objetos del navegador que hacen ambas versiones sean incompatibles con frecuencia. Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, ó Modelo de Objetos del Documento en castellano), que incorporan Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera la versión 7, y Mozilla Application Suite, Mozilla desde su primera versión.

Javascript puede incluirse en cualquier documento y es compatible con HTML en el navegador del cliente, ya sea PHP, Active Server Pages, ASP, JSP y SVG. Incluir código directamente en una estructura HTML es una práctica invasiva y no recomendada. El método correcto que define la W3C es incluir javascript como un archivo externo, tanto por cuestiones de accesibilidad, como práctica y velocidad en la navegación.

1.4.4 AJAX

Ajax, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Ajax es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

1.4.5 PHP

PHP es un acrónimo recursivo que significa Hypertext Preprocessor (Preprocesador de Hypertexto). Es un lenguaje de programación interpretado embebido en las páginas HTML. La mayoría de su sintaxis está basada en C, Java y Perl con algunas características propias. El principal objetivo del lenguaje es permitir a los desarrolladores de aplicaciones basadas en el web escribir páginas que se generan de forma dinámica de una forma sencilla y rápida. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida, lo cual aumenta la funcionalidad del servidor, convirtiéndolo en un sistema de desarrollo de aplicaciones cliente/servidor mucho más completo. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. A pesar de que es usado principalmente en interpretación del lado del servidor (server-side scripting), actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Un sitio con paginas dinámicas es el que permite interactuar con el visitante, de modo que cada usuario que visita la pagina vea la información modificada para requisitos articulares. Las aplicaciones dinámicas para el Web son frecuentes en los sitios comerciales e-commerce), donde el contenido visualizado se genera de la información alcanzada en una base de datos u otra fuente externa.

Una de sus características más potentes es su soporte para gran cantidad de bases de datos, entre las cuales pueden mencionarse MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite, InterBase, mSQL, Informix, Sybase, Solid, dBase, Adabas D, FilePro, Velocis, Unix dbm, entre otras

PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML.

Su sintaxis es muy similar a la del ASP, pues el código PHP va incrustado dentro del código HTML. Sus tags van incluidos dentro de `<? y ?>`.

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes.

Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

PHP es utilizado en aplicaciones Web-relacionadas por algunas de las organizaciones más prominentes tales como Mitsubishi, Redhat, Der Spiegel, MP3-Lycos, Ericsson, NASA, etc.

PHP es la opción natural para los programadores en máquinas con Linux que ejecutan servidores web con Apache, pero funciona bien en cualquier otra plataforma de UNIX o de Windows, Netscape o del web server de Microsoft. PHP también utiliza las sesiones de HTTP, conectividad de Java, expresiones regulares, LDAP, SNMP, IMAP, protocolos de COM (bajo Windows).

1.5 INGENIERÍA DE PROGRAMACIÓN

Ingeniería de software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como Desarrollo de Software o Producción de Software (Bohem, 1976).

Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software (IEEE, 1993).

Para el desarrollo de programas de cierta envergadura o complejos con ciertas garantías de calidad es conveniente seguir alguna de las metodologías de desarrollo de software.

1.5.1 METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Las metodologías en el desarrollo de software nos proporcionan las guías para poder conocer todo el camino a recorrer desde antes de empezar la implementación, con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado. Las metodologías de desarrollo de software se clasifican en tradicionales o convencionales y ágiles.

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.

Tabla 1.15 Diferencias entre metodologías ágiles y no ágiles.

Actualmente existen muchas metodologías para el desarrollo de software, por lo que a continuación se presenta un cuadro comparativo entre las metodologías más conocidas:

Modelo	Características	Ventajas	Desventajas
<i>Rational Unified Process (RUP)</i>	<p>Metodología tradicional o ágil. Es iterativo e incremental. Dirigido por los casos de uso. Centrado en la arquitectura. Desarrollo basado en componentes. UML como lenguaje de Modelado. Proceso Integrado. Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo). Incluye artefactos (que son los productos tangibles del proceso) y roles (papel que desempeña una persona en un determinado momento). Administración de requisitos. El cliente interactúa con el equipo de desarrollo mediante reuniones. Control de cambios. RUP se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición.</p>	<p>Es el proceso de desarrollo más general que existe. Se basa en las mejores prácticas que se han intentado y se han probado en el campo. Da énfasis en los requisitos y el diseño. Asegura la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos. Mitigación temprana de posibles riesgos altos. Progreso visible en las primeras etapas. Temprana retroalimentación que se ajuste a las necesidades reales. Gestión de la complejidad. Conocimiento adquirido en una iteración puede aplicarse de iteración a iteración.</p>	<p>Un cambio en las etapas de vida del sistema incrementaría notablemente el costo. La puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura. Es más apropiada para proyectos grandes (aunque también pequeños), dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios. Alta complejidad.</p>
<i>Extreme Programming (XP)</i>	<p>Metodología ágil. Para proyectos de corto plazo con un equipo pequeño y rápida entrega. El usuario final es parte del equipo. Refabricación.- Se basa en la utilización repetitiva de código. Programación en pares. Las fases de la metodología XP son: Planeación, Diseño, Codificación y Pruebas. Define 4 variables para proyectos de software: coste, tiempo, calidad y alcance.</p>	<p>Las iteraciones cortas permiten la retroalimentación. Se harán pruebas todo el tiempo, no sólo de cada nueva clase (pruebas unitarias) sino que también los clientes comprobarán que el proyecto va satisfaciendo los requisitos (pruebas funcionales). La participación del cliente en el equipo de trabajo. Flexibilidad al cambio.</p>	<p>El desarrollo de software es riesgoso y difícil de controlar. Se basa en prácticas inestables. El rol del cliente no refleja los diferentes intereses, habilidades y fuerzas a las que enfrentan los programadores durante el desarrollo de proyectos. Se trabaja exclusivamente en las necesidades del momento.</p>
<i>Cascada</i>	<p>Metodología tradicional. Orden riguroso de las etapas del ciclo de vida del software de tal forma que el inicio de una etapa debe esperar a la finalización de la etapa inmediatamente anterior. Cada fase genera documentación para la siguiente. Esta documentación debe ser aprobada. Requiere disponer de unos requisitos completos y precisos al principio del desarrollo. Todas las etapas deben desarrollarse para que el proyecto tenga éxito. Las fases continúan hasta que el objetivo es cumplido. Las fases del modelo en cascada son: Ingeniería y análisis del sistema, Análisis de requerimientos, Diseño, Codificación, Pruebas y Mantenimiento.</p>	<p>Ayuda a minimizar los gastos de la planificación porque permite realizarla sin problemas. Facilita la gestión del desarrollo. Evita una fuente común de errores importantes. Se utiliza correctamente para ciclos en los que se tiene una definición estable del producto.</p>	<p>Si se cambia el orden de las fases, el producto será de menor calidad. Se tarda mucho tiempo en pasar por todo el ciclo. Dificultad para especificar claramente los requerimientos al comienzo del proyecto (no permite flexibilidad en los cambios). Los errores de análisis y diseño son costosos de eliminar, y se propagan a las fases siguientes. Se genera mucho mantenimiento inicial debido al período de congelación de requisitos y éste recae, en su mayor parte, sobre el código fuente, que en consecuencia se va deteriorando y resultando cada vez más difícil de mantener.</p>
<i>SCRUM</i>	<p>Metodología ágil. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. Reuniones a lo largo proyecto, una reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. Para proyectos con un rápido cambio de requisitos.</p>	<p>Entrega de un producto funcional al finalizar cada Sprint. Posibilidad de ajustar la funcionalidad en base a la necesidad de negocio del cliente. Visualización del proyecto día a día. Alcance acotado y viable. Equipos integrados y comprometidos con el proyecto, toda vez que ellos definieron el alcance y se auto-administran.</p>	<p>No genera toda la evidencia o documentación de otras metodologías. No es apto para todos los proyectos. Tal vez sea necesario complementarlo con otros procesos (XP).</p>

Tabla 1.16 Comparación de metodologías de software (1 de 2).

Modelo	Características	Ventajas	Desventajas
<i>Espiral</i>	<p>Metodología tradicional.</p> <p>Las actividades de este modelo se conforman en una espiral, en la que cada iteración representa a un conjunto de actividades.</p> <p>Para cada ciclo del modelo en espiral hay cuatro actividades:</p> <ul style="list-style-type: none"> · Determinar o fijar objetivos. · Análisis del riesgo. · Planificar. · Desarrollar, verificar y validar (probar). <p>Toma en consideración explícitamente el riesgo, esta es una actividad importante en la administración del proyecto.</p>	<p>Reduce riesgos del proyecto.</p> <p>Integra el desarrollo con el mantenimiento.</p> <p>Utiliza las fases de modelos tradicionales.</p> <p>Se centra en la eliminación de errores y alternativas poco atractivas.</p> <p>Mejora al modelo en cascada enfatizando la naturaleza iterativa del proceso de diseño.</p> <p>Permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.</p> <p>Puede adaptarse y aplicarse a lo largo de la vida del software de computadora, no terminal cuando se entrega el software.</p>	<p>Genera mucho tiempo en el desarrollo del sistema.</p> <p>Costoso.</p> <p>Requiere experiencia en la identificación de riesgos y habilidad para saber cuándo detener la evolución.</p> <p>Consumo muchos recursos.</p> <p>Demstrar al cliente que el enfoque evolutivo es controlable.</p>
<i>Prototipos</i>	<p>Metodología tradicional.</p> <p>Se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado (en forma de un diseño rápido).</p> <p>El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. El diseño rápido conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.</p>	<p>Es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.</p> <p>También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.</p>	<p>El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final.</p> <p>A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte de los casos a reconstruirlo una vez que el prototipo ha cumplido su función.</p> <p>En aras de desarrollar rápidamente el prototipo, el desarrollador suele tomar algunas decisiones de implementación poco convenientes.</p>
<i>Adaptive Software Development (ASD)</i>	<p>Metodología ágil.</p> <p>Iterativo.</p> <p>Orientado a los componentes software más que a las tareas.</p> <p>Tolerante a los cambios.</p> <p>Guiado por los riesgos</p> <p>La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.</p> <p>El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje.</p>	<p>Utiliza información disponible acerca de cambios para mejorar el comportamiento del software.</p> <p>Promulga colaboración, la interacción de personas.</p>	<p>Usa los resultados de cada ciclo de desarrollo para adaptar el siguiente.</p> <p>Ambiente imprevisible que genera incertidumbre.</p>
<i>Incremental</i>	<p>Metodología tradicional.</p> <p>Es una combinación del Modelo de Cascada y Modelo Evolutivo.</p> <p>Es iterativo por naturaleza.</p> <p>Es un desarrollo inicial de la arquitectura completa del sistema, seguido de incrementos y versiones parciales del mismo. Cada incremento tiene su propio ciclo de vida y agrega funcionalidad adicional o mejorada sobre el sistema.</p> <p>Conforme se completa cada etapa, se verifica e integra la versión con las demás versiones ya completadas del sistema.</p> <p>En una visión genérica, el proceso se divide en cuatro partes: Análisis, Diseño, Código, Prueba. En este método se mantiene al cliente en constante contacto con los resultados obtenidos en cada incremento.</p>	<p>Los clientes no esperan hasta el fin del desarrollo para utilizar el sistema. Pueden empezar a usarlo desde el primer incremento.</p> <p>Los clientes pueden aclarar los requisitos que no tengan claros conforme ven las entregas del sistema.</p> <p>Se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento.</p> <p>Las partes más importantes del sistema son entregadas primero, por lo cual se realizan más pruebas en estos módulos y se disminuye el riesgo de fallos.</p> <p>Se puede financiar el proyecto por partes.</p>	<p>Cada incremento debe ser pequeño para limitar el riesgo.</p> <p>Cada incremento debe aumentar la funcionalidad.</p> <p>Es difícil establecer las correspondencias de los requisitos contra los incrementos.</p> <p>Es difícil detectar las unidades o servicios genéricos para todo el sistema.</p> <p>Hay costos ocultos en su implementación, ya que se incorporan varias actividades a realizar por el equipo.</p>

Tabla 1.16 Comparación de metodologías de software (2 de 2).

CAPÍTULO 2

ANÁLISIS DEL SISTEMA

2.1 DEFINICIÓN DEL PROBLEMA

NIC-UNAM necesita una base de datos con el objetivo de registrar y administrar la información de las redes de voz y datos de RedUNAM así como los datos de las dependencias e instituciones a quienes se les brinda el servicio y los datos personales de los administradores de red encargados de dichas redes en cada una de estas dependencias, entre otra información relevante concerniente a las redes.

RedUNAM está constituida de la siguiente manera: está formada por 4 nodos principales, los cuales son: DGSCA, IIMAS, ARQUITECTURA Y ZONA CULTURAL. En cada uno de ellos hay varios switches en los que se encuentran conectadas las redes y/o subredes mediante redes virtuales, o bien, de manera directa. Cabe mencionar que el segmento de red es lo que determina si se trata de una red o subred, de acuerdo al rango de direcciones IP que abarque ya que si éste va de 1 al 254, es un segmento completo, es decir, se trata de una red la cual es asignada a una sola dependencia; mientras que en todos los demás casos donde el segmento no está completo, se trata de subredes donde previamente hubo un subneteo de la red y por lo tanto pueden existir varias subredes de una misma red, o bien, de distintas redes, que estén asignadas a una o a diferentes dependencias. Por lo que en un switch y/o red virtual pueden estar conectadas varias dependencias y a su vez una dependencia puede estar en distintos switches y/o redes virtuales.

Los datos que se manejan de cada una de las dependencias son: nombre, clave, siglas, alias, tipo (interna/externa), conexión (si/no), dirección (calle y número, colonia, estado, delegación o municipio, código postal, ciudad y localidad) y un campo para observaciones. Una dependencia tiene uno o más dominios.

Las redes virtuales (VLANs) cuentan con una interfaz lógica caracterizada por un identificador y su nombre, dichas redes pueden tener configuradas una o muchas subredes. Para cada una de las redes se registran los siguientes datos: número de red (contador), tipo (datos/voz), identificador de red, máscara, gateway, rango de direcciones IP (1-254), nodo al que pertenece, switch en el que está conectada, red virtual, clasificación de red (LAN/WAN), interfaz física, enlaces y un campo para observaciones.

Existen enlaces, los cuales están asociados a una o a muchas redes WAN. Tales enlaces van hacia un sólo equipo, o bien, a equipos distintos. Para cada uno de ellos se registran los siguientes datos: identificador (contador), tributaria, time slot, puerto, DPI, bandwidth, ubicación y un campo para observaciones.

Cada dependencia tiene al menos un administrador encargado de la(s) red(es) y/o subred(es) asignada(s) a dicha institución, el cual funge como contacto técnico y/o administrativo. De ambos contactos se debe contar con la siguiente información: tipo

(técnico/administrativo), nombre completo (nombre(s), apellido paterno, apellido materno), clave, título, puesto, teléfono(s), celular, e-mail y un campo para observaciones. Un administrador puede estar a cargo de más de una dependencia.

De cada registro se necesita saber la fecha de creación (fecha en que se dio de alta) y la fecha de actualización (modificaciones).

2.2 ANÁLISIS DE REQUERIMIENTOS

Se requiere un sistema de bases de datos con las siguientes características:

✓ Seguridad

Los datos contenidos en la base de datos deberán estar protegidos de tal manera que sólo los usuarios autorizados tendrán acceso al sistema con ciertas restricciones o privilegios dependiendo del usuario del que se trate. Por lo tanto, para poder entrar al sistema, se requiere de un nombre de usuario y contraseña, los cuales serán validados, es decir, se buscará el registro previo del usuario en la base de datos y al encontrarse, el usuario podrá ingresar con los privilegios que le hayan sido otorgados, en caso contrario se negará el acceso. Para tal efecto, se definirán distintos tipos de usuario a los cuales se les asignarán permisos y restricciones. Además, en el servidor se guardarán las direcciones IP de las máquinas cliente autorizadas para que sólo desde ellas se puedan hacer consultas a la base de datos. Se asignará el tipo de cuenta a cada usuario de acuerdo a sus necesidades, lo cual implica que cierto número de personas pertenecerán a cada grupo.

La administración y mantenimiento del sistema estará designado a una sola persona o en su defecto, a un pequeño grupo de personas para que se pueda mantener un mayor y mejor control y una buena administración tanto del sistema como de los usuarios. El administrador tendrá el control total sobre el sistema y todos los permisos sobre la base. Asimismo, estará encargado de la creación de cuentas de los usuarios y también definirá los permisos a las tablas y archivos en general.

Las acciones realizadas en el sistema estarán registradas en una bitácora donde también estará el nombre del usuario que los llevó a cabo, así como la fecha y la hora correspondiente, con el objetivo de mantener la integridad de la información, y poder identificar a algún usuario para cualquier aclaración en los movimientos realizados.

✓ Integridad

Las reglas de negocio establecidas serán respetadas en las consultas hechas a la base de datos. De igual forma, se pondrá especial énfasis en la etapa de diseño para que en la fase lógica, las restricciones de integridad se mantengan, buscando siempre que los datos sean válidos y consistentes. El sistema gestor de la base de datos será relacional (RDBMS).

✓ Fiabilidad

Una vez que el sistema esté terminado, será sometido a un conjunto de pruebas para probar su resistencia ante fallos. Los errores encontrados serán corregidos y nuevamente se someterá a las pruebas y así sucesivamente hasta que sea necesario, de manera que se

erradiquen todos los errores posibles y poder garantizar el buen funcionamiento del sistema ya que éste debe estar disponible todo el tiempo, es decir, “7x24” (los 7 días de la semana, 24 horas cada día).

✓ **Eficiencia**

La consulta a la base de datos será en línea, es decir, vía web. El sistema es multiusuario y por lo tanto cada usuario sólo deberá contar con una computadora con acceso a Internet y un navegador web, sin importar su ubicación geográfica. Y los resultados de las consultas serán desplegados en cuestión de milésimas de segundos y así como ésta, las demás acciones (altas/bajas/modificaciones) ya que todo se ejecuta en tiempo real.

Y de esta forma se da solución a los problemas que se tienen actualmente, tales como:

- Gran consumo de recursos de la computadora, en especial de memoria RAM y espacio en disco duro.
- La consulta a la base de datos sólo puede ser hecha desde las computadoras que tienen instalado el software necesario para ello.
- La instalación de la aplicación para consultar la base de datos es diferente en cada sistema operativo con cierto grado de dificultad dependiendo de éste.

✓ **Portabilidad**

El software utilizado deberá ser totalmente portable, es decir, que el sistema pueda ejecutarse en diferentes plataformas sin ningún problema.

✓ **Flexibilidad**

La base de datos estará diseñada para poder adaptarse a cambios en los requerimientos, esto quedará plasmado desde el nivel más bajo de abstracción.

✓ **Escalabilidad**

El hecho que RedUNAM está en constante cambio y crecimiento se ha considerado como algo primordial, por lo que el sistema podrá crecer e implementar nuevos requerimientos para así poder satisfacer las necesidades que se vayan presentando. Todo esto se tomará en cuenta en las distintas etapas de la metodología del desarrollo de software.

✓ **Interoperabilidad**

La arquitectura y el sistema operativo del servidor y de los clientes pueden ser distintos, esto es, es SW y HW no intervienen en la interacción entre ellos ni en su desempeño.

✓ **Facilidad de uso**

Se brindará al usuario pantallas donde pueda ubicar fácilmente lo que necesite, procurando que realice el menor número de selecciones durante su visita al sitio.

✓ Interfaz amigable

Los mensajes o avisos que el sistema presente al usuario serán claros y precisos. En algunos casos contarán con ayuda visual por medio de colores que indicarán al usuario una acción a realizar.

✓ Bajo costo

Se utilizará software libre en el desarrollo del sistema, por lo que en realidad éste no tendrá ningún costo, no se tendrán que adquirir licencias ni tampoco ningún tipo de software especial.

2.3 ALCANCES

El sistema proporcionará vía web (en línea) los servicios de altas, bajas, cambios y consultas a los usuarios del Centro de Operación de la Red acerca de los registros de las redes de voz y datos de RedUNAM administradas por NIC-UNAM, las dependencias a las que fueron asignadas dichas redes y los administradores a cargo de ellas, basándose en el esquema actual de los DNS de RedUNAM (Figura 2.1), incluyendo nodos, switches, redes virtuales, enlaces y equipos; Asimismo, contará con la parte concerniente a la administración de los usuarios donde se podrán dar de alta los usuarios y grupos correspondientes, entre otras funciones que tienen que ver con éstos. Además contará con una bitácora de los usuarios que entren al sistema y las acciones que realizaron en él con la fecha y hora correspondiente y se cubrirán los requerimientos previamente mencionados. Se utilizará la metodología RUP para el desarrollo de este proyecto.

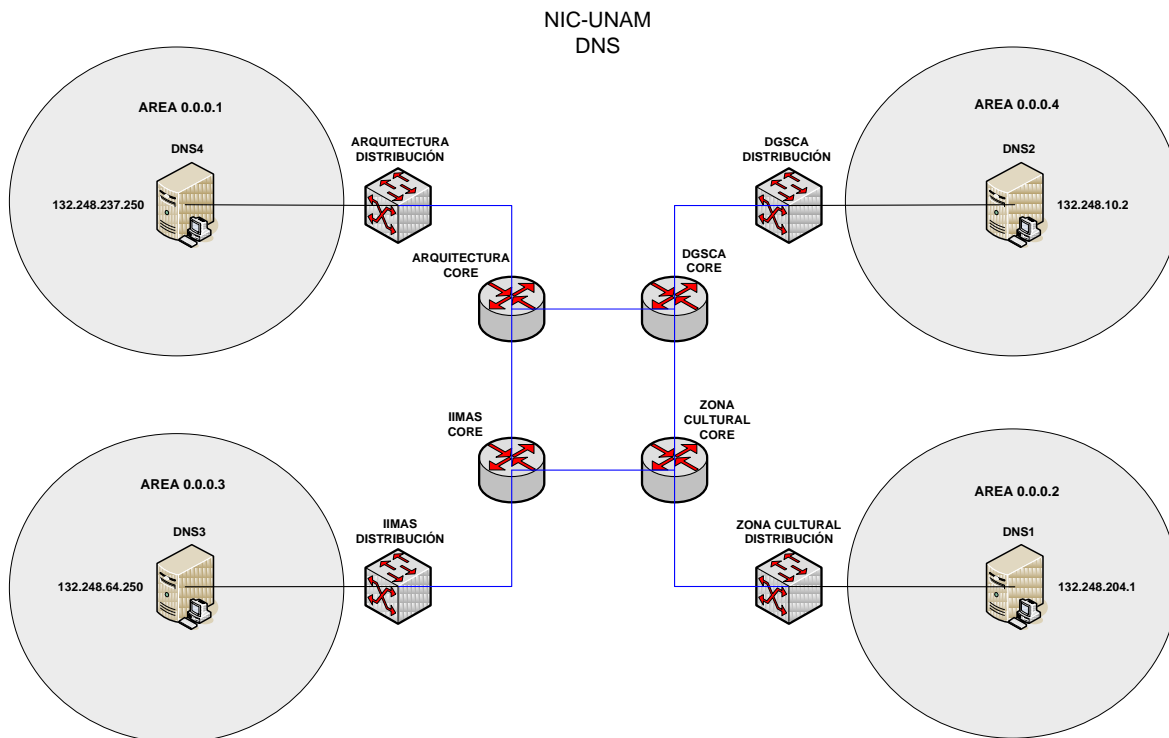


Figura 2.1 Esquema actual de los DNS de RedUNAM.

2.4. CASOS DE USO

Introducción

Diagrama de casos de uso

Un diagrama de casos de uso muestra, los requerimientos funcionales que se esperan de una aplicación o sistema y el comportamiento derivado de la interacción con su entorno (usuarios u otras aplicaciones o sistemas).

Los elementos básicos de un diagrama de casos de uso son el sistema, los actores, los casos de uso, las asociaciones y las relaciones.

Sistema

Define los límites entre el sistema a desarrollar y los elementos externos a ese sistema (actores usuarios del sistema).

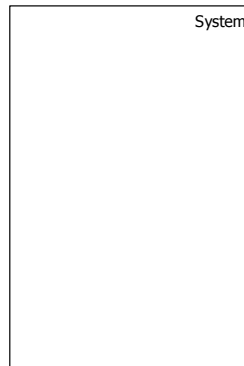


Figura 2.2 Representación de un sistema.

Actores

Se denomina actor a toda entidad externa (personas, sistemas, máquinas o abstractas) al sistema que guarda una relación con éste y que le demanda una funcionalidad.

Es importante tener clara la diferencia entre usuario y actor. Un actor es una clase de rol, mientras que un usuario es una persona que, cuando usa el sistema, asume un rol. De esta forma, un usuario puede acceder al sistema como distintos actores.

Los actores se representan con dibujos simplificados de personas, llamados en inglés “stick man” (hombres de palo) y su respectivo nombre al pie de la figura:

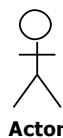


Figura 2.3 Representación de un actor.

Casos de uso

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software.

Los casos de uso tienen las siguientes características:

- Están expresados desde el punto de vista del actor.
- Se documentan con texto informal.
- Describen tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él, aunque el énfasis está puesto en la interacción.
- Son iniciados por un único actor.
- Están acotados al uso de una determinada funcionalidad claramente diferenciada del sistema.
- Son nombrados usando una frase (verbo), expresando la meta que debe cumplir el sistema.

Un caso de uso tiene como instancias los escenarios, los cuales son situaciones concretas que deben recorrer total o parcialmente el caso de uso. Se deben considerar en lo posible todos los escenarios de modo que se pueda validar el caso de uso. La última comprobación consiste por tanto en asegurar que el caso de uso represente todos los escenarios, es decir sea una generalización de ellos.

Un caso de uso se representa con una elipse y el nombre del caso en su interior.



Figura 2.4 Representación de un caso de uso.

Asociaciones

Hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso. Se representan con una línea conectando un actor a un caso de uso. Pueden ser bidireccionales o unidireccionales.

Relaciones

Existen tres tipos de relaciones en los diagramas de casos de uso:

1. Incluye (<<include>>).- Es un estereotipo de dependencia. Indica que un caso de uso es incluido en otro. Típicamente ocurre cuando unos casos de uso comparten unos pasos en común. El caso de uso incluido es el factor común del comportamiento compartido.

2. Generalización (sin estereotipo).- Indica que un caso de uso es una variante de otro. El caso de uso especializado puede variar cualquier aspecto del caso de uso base. Se refiere a relaciones de herencia para mostrar que diferentes elementos están relacionados como tipos de otros. Son aplicables a casos de uso o actores.
3. Extiende (<<extend>>).- Es un estereotipo de dependencia. Ofrece una forma de extensión más controlada que la relación de generalización. El caso de uso base declara un conjunto de puntos de extensión. El caso de uso especializado sólo puede alterar el comportamiento de los puntos de extensión marcados.

Relación	Función	Notación
<i>Asociación</i>	Camino de comunicación entre un actor y una caso de uso en el que participa	—————
<i>Extiende</i>	Inserción de comportamiento adicional en un caso de uso base (<i>sin que éste tenga conocimiento</i>)	-----<<extiende>>→
<i>Generalización</i>	Relación entre un caso de uso general y otro más específico que hereda características y añade otras	—————▷
<i>Incluye</i>	Inserción de comportamiento adicional dentro de un caso de uso que explícitamente describe la inserción	-----<<incluye>>→

Figura 2.5 Tipos de relaciones.

2.4.1 REQUERIMIENTOS FUNCIONALES DEL SISTEMA DE BASES DE DATOS

Actores

En el sistema participan tres tipos de actores, los cuales son: usuario de consulta, usuario de edición y administrador. Los cuales desempeñan los siguientes roles:

Usuario de consulta.- Es aquel que entra al sistema con el fin de obtener información específica contenida en la base de datos. Sólo tiene permiso de lectura, es decir, sólo puede consultar la información de la base sin modificarla.

Usuario de edición.- Este usuario además de consultar la información, puede modificar los datos contenidos en la base mediante la edición de éstos, o bien, insertar nueva información agregando nuevos registros. También podrá consultar la bitácora del sistema.

Administrador.- El administrador tiene todos los privilegios sobre la base de datos: consultar, agregar, modificar y eliminar la información contenida en ella; además, está a cargo de la seguridad, del monitoreo de la bitácora y de la creación de usuarios con sus

respectivas contraseñas las cuales proporcionará a las personas que tendrán acceso. Asimismo, posee todo el control sobre el sistema en cuanto a diseño y estructura se refiere de tal forma que cualquier cambio interno o externo será realizado por él.

En la siguiente figura se presentan los casos de uso del sistema:

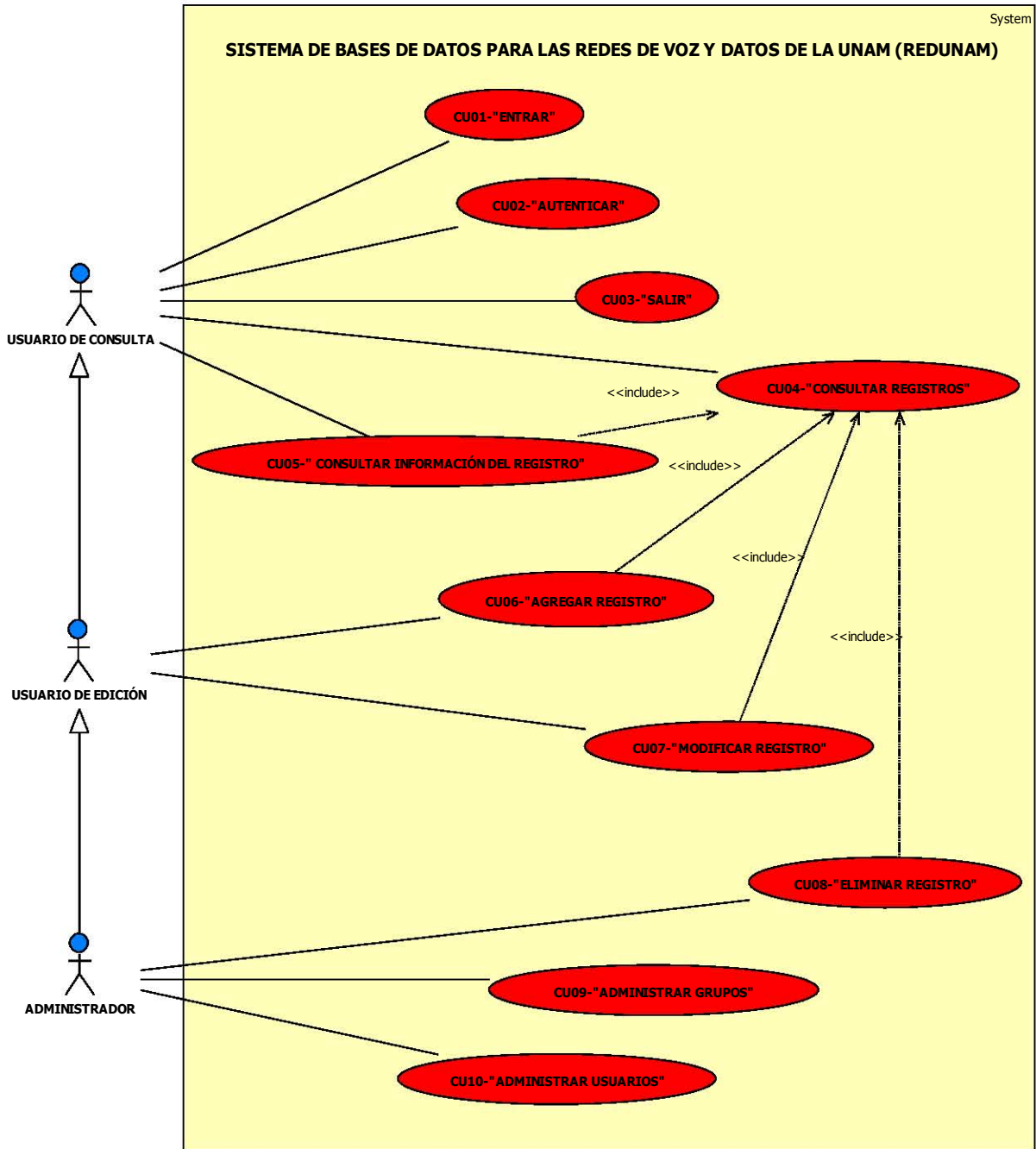


Figura 2.6 Diagrama general de casos de uso.

A continuación se presenta la descripción detallada de cada uno de los casos de uso.

❖ **ESPECIFICACIONES DE LOS CASOS DE USO**

CASO DE USO		"<u>ENTRAR</u>"
No. CU		<u>01</u>
ACTORES		Usuario de consulta, usuario de edición y administrador.
DESCRIPCIÓN		Permite a los usuarios entrar al sitio web del sistema de bases de datos de RedUNAM escribiendo la dirección URL en un navegador.
PRE-CONDICIONES		<p>Del proceso</p> <ul style="list-style-type: none"> • El usuario deberá conocer la URL del sistema de bases de datos. • El usuario debe estar interesado en obtener información concerniente a las redes de RedUNAM administradas por NIC-UNAM. <p>Equipo requerido:</p> <ul style="list-style-type: none"> • Una computadora que tenga instalado un navegador Web (I. Explorer, Mozilla, Opera, etc.). • Conexión a la Internet.
FLUJO PRINCIPAL		
Paso	Acción	Excepción
1	El usuario escribe en el navegador la URL del sistema de bases de datos de RedUNAM.	
2	Se despliega la página principal del sistema.	E1
EXCEPCIONES		
E	Tipo	Acción
1	Dirección incorrecta	El navegador despliega un mensaje de error 404, es decir, la página no existe.
POST-CONDICIONES		Opción de entrar al Sistema mediante el ingreso de nombre de usuario y la contraseña correspondiente en la página principal.

CASO DE USO		<u>"AUTENTICAR"</u>	
No. CU		<u>02</u>	
ACTORES		Usuario de consulta, usuario de edición y administrador.	
DESCRIPCIÓN		Permite el acceso de los usuarios al sistema de bases de datos de RedUNAM mediante la verificación del nombre de usuario y contraseña.	
PRE-CONDICIONES		Del proceso: <ul style="list-style-type: none"> • El usuario debe estar previamente registrado en el sistema de bases de datos de RedUNAM. • El usuario ingresó al sistema de bases de datos de RedUNAM (CU_01). 	
FLUJO PRINCIPAL			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	En la pantalla principal del Sistema, el usuario ingresa su nombre de usuario y contraseña y selecciona "Entrar".		
2	El sistema confronta la información proporcionada por el usuario contra la información de los usuarios autorizados registrados en el sistema. Se presenta la página de inicio.		E1, E2 ó E3
EXCEPCIONES			
<i>E</i>	<i>Tipo</i>	<i>Acción</i>	
1	Nombre de usuario y contraseña obligatorios.	En caso de faltar uno de ellos se mostrará la leyenda "Nombre de usuario obligatorio"/"Contraseña obligatoria" dependiendo del campo del que se trate. Se vuelve a mostrar la pantalla principal.	
2	Nombre de usuario y/o contraseña no válidos.	En caso de que el usuario y/o la contraseña proporcionados no estén previamente registrados como usuarios autorizados en el sistema, se mostrará el siguiente mensaje: "Nombre de usuario y/o contraseña no válidos". Se muestra nuevamente la pantalla principal.	
3	Sesión abierta	En caso de que el usuario ya tenga una sesión abierta se lanzará el siguiente mensaje: "El usuario ya tiene una sesión abierta". Se muestra la pantalla principal con la opción de volver a introducir la contraseña, o bien, iniciar con otra cuenta.	
POST-CONDICIONES		El usuario tendrá acceso a sus respectivas funciones de acuerdo al tipo de usuario del que se trate.	

CASO DE USO		"SALIR"	
No. CU		<u>03</u>	
ACTORES		Usuario de consulta, usuario de edición y administrador.	
DESCRIPCIÓN		Permite al usuario salir del sitio.	
PRE-CONDICIONES		Del proceso <ul style="list-style-type: none"> • El usuario ingresó al sistema de bases de datos (CU_01). • El usuario fue autorizado (CU_02). 	
FLUJO PRINCIPAL			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	El usuario elige salir del sistema mediante: <ul style="list-style-type: none"> 1.1 Cerrar el navegador. 1.2 Cerrar sesión. 1.3 Cambiar de página. 		
2	El usuario sale del Sistema de bases de datos.		
FLUJOS ALTERNOS			
"1.1 CERRAR EL NAVEGADOR"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1.1.1	El usuario cierra la ventana de trabajo del navegador.		
1.1.2	Se cierra el navegador sin cerrar la sesión del usuario. Ir a paso 2.		
"1.2 CERRAR SESIÓN"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1.2.1	El usuario selecciona la opción "Cerrar sesión" del sistema.		
1.2.2	Se cierra la sesión del usuario y se despliega la página principal del Sistema. Ir a paso 2.		
"1.3 CAMBIAR DE PÁGINA"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1.3.1	El usuario teclea otra URL en el navegador.		
1.3.2	Se despliega la nueva página sin cerrar sesión. Ir a paso 2.		E1
EXCEPCIONES			
<i>E</i>	<i>Tipo</i>	<i>Acción</i>	
1	Dirección incorrecta	El navegador despliega un mensaje de error 404, es decir, la página no existe.	
POST-CONDICIONES		<ul style="list-style-type: none"> • Sesión abierta para los usuarios que cerraron el navegador o cambiaron de página. • Opción de volver a entrar al sistema para los usuarios que cerraron sesión. 	

CASO DE USO		"CONSULTAR REGISTROS"	
No. CU		<u>04</u>	
ACTORES		Usuario de consulta, usuario de edición y administrador.	
DESCRIPCIÓN		Brinda información al usuario acerca de los registros existentes en la base de datos de RedUNAM de acuerdo a una consulta realizada por éste.	
PRE-CONDICIONES		Del proceso: <ul style="list-style-type: none"> • El usuario ingresó al sistema de bases de datos de RedUNAM (CU_01). • El usuario fue autorizado (CU_02). 	
FLUJO PRINCIPAL			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	El usuario selecciona una opción del menú principal.		
2	El sistema despliega todos los registros que se tienen en la base de datos correspondientes a la opción seleccionada previamente.		E1, E2
3	El usuario tiene dos opciones de búsqueda: 3.1 Búsqueda general. 3.2 Búsqueda avanzada.		
4	Las coincidencias encontradas entre la petición y la información registrada en el sistema son desplegadas en forma de lista.		E1, E2
FLUJOS ALTERNOS			
"3.1 BÚSQUEDA GENERAL"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
3.1.1	El usuario realiza una búsqueda general.		
3.1.2	El usuario introduce la palabra clave que desee consultar en el área de texto correspondiente.		
3.1.3	El usuario selecciona "Buscar".		
3.1.4	El sistema procede a realizar la búsqueda correspondiente. Ir a paso 4.		
"3.2 BÚSQUEDA AVANZADA"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
3.2.1	El usuario realiza una búsqueda avanzada.		
3.2.2	El usuario elige uno de los registros desplegados por el sistema, seleccionando cualquier columna excepto las columnas editar y borrar.		
3.2.3	El sistema abre una o más pestañas de acuerdo al registro seleccionado.		E1, E2
3.2.4	El usuario puede: <ul style="list-style-type: none"> 3.2.3.1 Ir a paso 3.2.2, en este caso la búsqueda será cada vez más específica logrando llegar hasta el nivel que el usuario desee y que el sistema le permita. 3.2.3.2 Ir a paso 1. 		
3.2.5	Ir a paso 4.		

CASO DE USO		<u>“CONSULTAR REGISTROS”</u>
No. CU		<u>04</u>
EXCEPCIONES		
<i>E</i>	<i>Tipo</i>	<i>Acción</i>
1	Problemas en el Sistema.	Si existe un problema en la red u otro referente a la base de datos, no podrá desplegarse la pantalla correspondiente a la selección y se mostrará un mensaje de “ERROR, contacte a su administrador”.
2	No hay registro.	La información solicitada no está registrada, es decir, no se encontró ningún dato en la base de datos que coincida con la solicitud hecha. El Sistema mandará el siguiente mensaje: "No se encontraron registros".
POST-CONDICIONES		El usuario puede: <ul style="list-style-type: none"> • Consultar más registros (CU04). • Consultar (CU05), agregar (CU06), modificar (CU07) o eliminar algún registro (CU08). • Salir del sistema (CU03). • Ir a la página de inicio.

CASO DE USO	“CONSULTAR INFORMACIÓN DEL REGISTRO”	
No. CU	<u>05</u>	
ACTORES	Usuario de consulta, usuario de edición y administrador.	
DESCRIPCIÓN	Permite al usuario consultar información de su interés acerca de RedUNAM contenida en la base de datos, mediante los registros.	
PRE-CONDICIONES	Del proceso: <ul style="list-style-type: none"> • El usuario ingresó al sistema de bases de datos (CU_01) • El usuario fue autorizado (CU_02). • El usuario realizó una búsqueda (CU_04). 	
FLUJO PRINCIPAL		
Paso	Acción	Excepción
1	El usuario selecciona uno de los registros de la lista desplegada por el sistema de acuerdo a la búsqueda previamente realizada ya sea general o avanzada.	
2	El usuario selecciona cualquier columna excepto las columnas editar y borrar.	
3	El sistema despliega la información del registro seleccionado, mediante una o más pestañas, según sea el caso.	E1
EXCEPCIONES		
E	Tipo	Acción
1	Información no disponible temporalmente.	No se mostrará la información solicitada en caso de que ésta esté siendo modificada en ese momento. El mensaje que se enviará al usuario será: "Información no disponible temporalmente. Intente más tarde por favor".
POST-CONDICIONES	El usuario puede: <ul style="list-style-type: none"> • Realizar una nueva búsqueda (CU04). • Consultar (CU05), agregar (CU06), modificar (CU07) o eliminar un registro (CU08). • Salir del sistema (CU03). • Ir a la página de inicio. 	

CASO DE USO		"AGREGAR REGISTRO"
No. CU		<u>06</u>
ACTORES		Usuario de edición y administrador.
DESCRIPCIÓN		Permite al usuario ingresar información a la base de datos mediante el llenado del formulario correspondiente.
PRE-CONDICIONES		Del proceso: <ul style="list-style-type: none"> • El usuario ingresó al Sistema de bases de datos (CU01) • El usuario fue autorizado (CU02). • El usuario consultó registros del sistema (CU04).
FLUJO PRINCIPAL		
<i>Paso</i>	<i>Acción</i>	<i>Excepción</i>
1	El usuario selecciona una de las pestañas con la información correspondiente a la consulta de registros.	
2	El sistema abre la página de la pestaña seleccionada.	E1
2	El usuario selecciona agregar “campo”, el cual puede ser uno de los siguientes debido al paso 1: <ul style="list-style-type: none"> • Dependencia (A-1).- Permite la captura de los datos de la dependencia. • Contacto (A-2).- Permite la captura de los datos personales del contacto técnico y/o administrativo. • Red (A-3).- Permite la captura de los datos correspondientes a las redes. • Equipo (A-4).- Permite la captura de información correspondiente a los equipos. • Nodo (A-5).- Permite la captura de los datos correspondientes a los nodos. • Switch (A-6).- Permite la captura de los datos correspondientes a los switches. • VLAN (A-7).- Permite la captura de los datos correspondientes a las redes virtuales. • Dominio (A-8).- Permite la captura de los datos correspondientes a los dominios. • Contacto a una Dependencia (A-9).- Permite agregar un contacto a una dependencia. • Enlace a un Equipo (A-10).- Permite agregar un enlace a un equipo. 	
3	El sistema despliega los campos a ser llenados en la misma página para introducir en cada uno de ellos la información correspondiente.	E1
4	Después de haber introducido todos los datos correspondientes en los formularios (A-1, A-2, A-3, A-4, A-5, A-6, A-7, A-8, A-9 o A-10). El usuario selecciona “Agregar” para enviar la información a la base de datos, o bien, “Cancelar”.	
5	Se envían los datos a la base de datos, éstos son almacenados y se actualiza el sistema.	E4, E5

CASO DE USO		"AGREGAR REGISTRO"	
No. CU		<u>06</u>	
FLUJOS ALTERNOS			
"A-1 DEPENDENCIA"			
Paso	Acción		Excepción
1	<p>El usuario introduce los siguientes datos de la dependencia:</p> <ul style="list-style-type: none"> • Nombre.- Se refiere al nombre de la dependencia. • NIC-HANDLE Dependencia.- Clave única de la dependencia asignada por el administrador del Sistema. • Siglas.- Iniciales de la dependencia. • Alias.- Sobrenombre o abreviatura de la dependencia. • Tipo.- Es el tipo de institución y hay dos tipos que son: <ul style="list-style-type: none"> - Interna: pertenece a la UNAM. - Externa: no pertenece a la UNAM. • Conexión.- Se refiere a la conexión a RedUNAM. El hecho de estar conectada significa que si se le está proporcionando el servicio, en caso contrario, significa que alguna vez estuvo conectada pero actualmente no se le brinda el servicio. • Dirección.- Es la dirección física de la institución integrada por calle y número, colonia, delegación o municipio, código postal, ciudad, estado, localidad. • Observaciones.- Alguna aclaración acerca de los campos previamente mencionados, o algo que se desee agregar acerca de la dependencia. <p>Ir a paso 4.</p>		E2, E3
"A-2 CONTACTO"			
Paso	Acción		Excepción
1	<p>El usuario introduce la siguiente información del contacto:</p> <ul style="list-style-type: none"> • Nombre.- El nombre completo del administrador: nombre(s), apellido paterno y apellido materno. • NIC-HANDLE Contacto.- Clave única del administrador de red asignada por el administrador del sistema. • Título.- Título profesional. • Puesto.- Cargo que ocupa dentro de la institución. • Teléfono.- Número(s) telefónico(s) donde pueda ser localizado, generalmente de la oficina donde labora. • Celular.- Número telefónico personal. • Correo electrónico.- dirección de correo electrónico de cualquier índole donde pueda enviarse información y así mantenerse en contacto con NIC-UNAM. • Observaciones.- Alguna aclaración acerca de los campos previamente mencionados, o algo que se desee agregar acerca del contacto. <p>Ir a paso 4.</p>		E2, E3

CASO DE USO		"AGREGAR REGISTRO"	
No. CU		<u>06</u>	
FLUJOS ALTERNOS			
"A-3 RED"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	<p>El usuario introduce la siguiente información acerca de la red:</p> <ul style="list-style-type: none"> • Tipo de red.- Se refiere a qué tipo de red es la que se está registrando, éstas pueden ser de 2 tipos distintos: datos o voz. • Identificador de red.- Es el número que identifica a una red. • Máscara.- Máscara de red aplicada. • Gateway.- Puerta de enlace o de acceso a otra red. • Rango de red.- Un conjunto de direcciones IP de una red que va del 1 al 254. • Clasificación de red.- Clasificación a la que pertenece dicha red de acuerdo a la cobertura geográfica que abarca, puede ser LAN o WAN. • Interfaz física.- Es la interfaz obligatoria para cualquier red clasificada de la siguiente manera: V35, G703, Ethernet (UTP, F.O.), FastEthernet (UTP, F.O.), GigabitEthernet (UTP, F.O.) y 10GigabitEthernet UTP. • Observaciones.- Alguna aclaración acerca de los campos previamente mencionados, o algo que se desee agregar acerca de la red. <p>Ir a paso 4.</p>		E2, E3
"A-4 EQUIPO"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	<p>El usuario ingresa la siguiente información acerca del equipo:</p> <ul style="list-style-type: none"> • Tributaria.- Dato de Telmex de donde está entregando el enlace. • Time slot.- Canal(es) configurado(s) para el enlace. • Puerto.- Interfaz donde se encuentra conectado el enlace. • DPI.- Referencia que Telmex da para identificar cada enlace. • Bandwidth.- Ancho de banda asignado al enlace. • Ubicación.- Ruteador donde se encuentra conectado. • Observaciones.- Alguna aclaración acerca de los campos previamente mencionados, o algo que se desee agregar acerca del equipo. <p>Ir a paso 4.</p>		E2, E3

CASO DE USO		"AGREGAR REGISTRO"	
No. CU		<u>06</u>	
FLUJOS ALTERNOS			
"A-5 NODO"			
Paso	Acción		Excepción
1	El usuario ingresa la siguiente información acerca del nodo: <ul style="list-style-type: none"> Nombre del Nodo.- Se refiere al nombre con que es identificado el nodo. Ir a paso 4.		E2, E3
"A-6 SWITCH"			
Paso	Acción		Excepción
1	El usuario ingresa la siguiente información acerca del switch: <ul style="list-style-type: none"> Nombre del Switch.- Se refiere al nombre con que es identificado el switch. Ir a paso 4.		E2, E3
"A-7 VLAN"			
Paso	Acción		Excepción
1	El usuario ingresa la siguiente información acerca de la red virtual: <ul style="list-style-type: none"> Nombre de la VLAN.- Se refiere al nombre con que es identificada la red virtual. Ir a paso 4.		E2, E3
"A-8 DOMINIO"			
Paso	Acción		Excepción
1	El usuario ingresa la siguiente información acerca del dominio: <ul style="list-style-type: none"> Nombre del dominio.- Se refiere al nombre asociado a una dirección IP. Ir a paso 4.		E2, E3
"A-9 CONTACTO A UNA DEPENDENCIA"			
Paso	Acción		Excepción
1	El usuario ingresa la siguiente información acerca del contacto: <p>Primero selecciona al contacto que quiere agregar y posteriormente agrega el tipo de contacto que será.</p> <ul style="list-style-type: none"> Tipo de Contacto.- Se refiere al tipo de contacto, existen dos tipos que son: <ul style="list-style-type: none"> Técnico.- Administrador de la red en el aspecto físico. Administrativo.- Administrador de la red y titular en el aspecto administrativo. Ir a paso 4.		E2, E3
"A-10 ENLACE A UN EQUIPO"			
Paso	Acción		Excepción
1	El usuario selecciona el enlace al que quiere agregar un equipo y posteriormente elige el equipo mediante: <ul style="list-style-type: none"> Ubicación del Equipo. Ir a paso 4.		E2, E3

CASO DE USO		"AGREGAR REGISTRO"
No. CU		<u>06</u>
EXCEPCIONES		
<i>E</i>	<i>Tipo</i>	<i>Acción</i>
1	Problemas en el Sistema.	Si existe un problema en la red u otro referente a la base de datos, no podrá desplegarse la pantalla correspondiente a la selección y se mostrará un mensaje de "ERROR, contacte a su administrador".
2	Información incompleta.	Falta llenar algunos campos que son de carácter obligatorio. Aparecerá el siguiente mensaje "Campo obligatorio".
3	Datos incongruentes.	Los datos introducidos no corresponden al tipo de dato asignado para ese campo. Se avisa al usuario mediante el siguiente mensaje: "Dato incorrecto" en el campo correspondiente.
4	Registro no completado.	Si existe un problema con la conexión u otro referente a la base de datos. No se podrá insertar el registro en ella, por lo que se lanzará un mensaje de "ERROR en la base de datos, no se pudo completar el registro".
5	Registro ya existente.	Si el sistema ya tiene registrada la información, se le notificará al usuario mediante el siguiente mensaje: "Esta (información) ya ha sido agregada".
POST-CONDICIONES		El usuario puede: <ul style="list-style-type: none"> • Agregar un nuevo registro (CU06). • Salir del sistema (CU03). • Ir a la página de inicio.

CASO DE USO		"MODIFICAR REGISTRO"	
No. CU		<u>07</u>	
ACTORES		Usuario de edición y administrador.	
DESCRIPCIÓN		Permite al usuario realizar modificaciones de la información previamente registrada en la base de datos mediante la actualización de los datos.	
PRE-CONDICIONES		Del proceso: <ul style="list-style-type: none"> • El usuario ingresó al sistema de bases de datos (CU_01) • El usuario fue autorizado (CU_02). • El usuario realizó una consulta mediante una búsqueda (CU_04). 	
FLUJO PRINCIPAL			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	El usuario selecciona uno de los registros de la lista desplegada por el sistema de acuerdo a la búsqueda previamente realizada ya sea general o avanzada.		
2	El usuario elige la opción "Editar".		
3	Se despliega la pantalla de edición en una nueva pestaña.		E1
4	El usuario realiza los cambios necesarios en los campos correspondientes y selecciona "Actualizar".		E2
5	El Sistema almacena/actualiza la información mostrando un mensaje de éxito de la transacción y cierra la pestaña de edición.		E3
EXCEPCIONES			
<i>E</i>	<i>Tipo</i>	<i>Acción</i>	
1	Problemas en el sistema.	Si existe un problema en la red u otro referente a la base de datos, no podrá desplegarse la pantalla correspondiente a la selección y se mostrará un mensaje de "ERROR, contacte a su administrador".	
2	Suspensión temporal.	No se podrá realizar ningún cambio si otro usuario está editando la información en ese momento. Por lo que se enviará el siguiente mensaje al usuario: "No procede. Intente más tarde por favor".	
3	Actualización falló.	En caso de que la actualización no pueda realizarse con éxito en la base de datos, el sistema lanzará el siguiente mensaje en la pantalla: "ERROR, la actualización no pudo ser realizada con éxito".	
POST-CONDICIONES		El usuario puede: <ul style="list-style-type: none"> • Realizar una nueva búsqueda (CU04). • Consultar (CU05), agregar (CU06), modificar (CU07) o eliminar un registro (CU08). • Salir del sistema (CU03). • Ir a la página de inicio. 	

CASO DE USO		“ELIMINAR REGISTRO”	
No. CU		<u>08</u>	
ACTORES		Administrador.	
DESCRIPCIÓN		Permite al usuario eliminar información de la base de datos.	
PRE-CONDICIONES		Del proceso: <ul style="list-style-type: none"> • El usuario ingresó al sistema de bases de datos (CU01) • El usuario fue autorizado (CU02). • El usuario realizó una consulta (CU04). 	
FLUJO PRINCIPAL			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	El usuario selecciona uno de los registros de la lista desplegada por el sistema de acuerdo a la búsqueda previamente realizada ya sea general o avanzada.		
2	El usuario elige la opción “Borrar”.		
3	El Sistema despliega el siguiente mensaje: ¿Está seguro de borrar “el registro”? Aceptar /Cancelar”.		
4	El usuario procede con la eliminación, es decir, elige “Aceptar”.		E1
5	Se elimina la información de la base de datos y se actualiza el sistema.		E2
EXCEPCIONES			
<i>E</i>	<i>Tipo</i>	<i>Acción</i>	
1	Eliminación cancelada.	En caso de elegir “Cancelar”, el sistema no procede con la petición anteriormente hecha.	
2	Eliminación falló.	Si el Sistema no puede llevar a cabo la eliminación de la información ya sea de forma parcial o total, se lanzará el siguiente mensaje “ERROR, no se ha podido completar la eliminación”.	
POST-CONDICIONES		El usuario puede: <ul style="list-style-type: none"> • Realizar una nueva búsqueda (CU04). • Consultar (CU05), agregar (CU08), modificar (CU07) o eliminar un registro (CU08). • Salir del sistema (CU03). • Ir a la página de inicio. 	

CASO DE USO		"ADMINISTRAR GRUPOS"	
No. CU		<u>09</u>	
ACTORES		Administrador.	
DESCRIPCIÓN		Permite al usuario administrar los grupos de usuarios del sistema mediante altas, bajas, cambios y consultas de éstos.	
PRE-CONDICIONES		Del proceso: <ul style="list-style-type: none"> • El usuario ingresó al sistema de bases de datos de RedUNAM (CU_01). • El usuario fue autorizado (CU_02). 	
FLUJO PRINCIPAL			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	El usuario selecciona la opción del menú principal concerniente a los grupos.		
2	El sistema despliega todos los registros existentes correspondientes a los grupos de usuarios.		E1, E2
3	El usuario puede realizar: <ul style="list-style-type: none"> 3.1 Consulta avanzada. 3.2 Editar grupos. 3.3 Borrar grupos. 3.4 Agregar grupos. 		
4	El sistema se actualiza y muestra todos los registros existentes correspondientes a los grupos de usuarios.		E5, E7
FLUJOS ALTERNOS			
"3.1 CONSULTA AVANZADA"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
3.1.1	El usuario elige uno de los registros desplegados por el sistema, seleccionando cualquier columna excepto las columnas editar y borrar.		
3.1.2	El sistema abre otra pestaña y en ella se despliega la información correspondiente al registro seleccionado.		E1, E3
3.1.3	El usuario puede configurar los permisos mediante la selección de la columna "Permisos", o ir a paso 4.		E6
"3.2 EDITAR GRUPOS"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
3.2.1	El usuario elige uno de los registros desplegados por el sistema, seleccionando "Editar".		
3.2.2	Se despliega la pantalla de edición.		E1
3.2.3	El usuario realiza los cambios necesarios en los campos correspondientes y selecciona "Actualizar". Ir a paso 4.		E4, E6, E8, E9

CASO DE USO		"ADMINISTRAR GRUPOS"	
No. CU		<u>09</u>	
FLUJOS ALTERNOS			
"3.3 BORRAR GRUPOS"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
3.3.1	El usuario elige uno de los registros desplegados por el sistema, seleccionando "Borrar".		
3.3.2	El Sistema despliega el siguiente mensaje: ¿Está seguro de borrar "el registro"? Aceptar /Cancelar".		E1
3.3.3	El usuario procede con la eliminación, es decir, elige "Aceptar".		E6
3.3.4	Se elimina la información de la base de datos. Ir a paso 4.		
"3.4 AGREGAR GRUPOS"			
3.4.1	El usuario selecciona agregar nuevo grupo.		
3.4.2	Se despliega el campo a ser llenado, en este caso sólo el nombre del grupo.		E1
3.4.3	El usuario introduce el nombre del grupo y selecciona agregar.		E6, E8, E9
3.4.4	El sistema envía la información a la base de datos y se almacenan los datos. Ir a paso 4		E10, E11
EXCEPCIONES			
<i>E</i>	<i>Tipo</i>	<i>Acción</i>	
1	Problemas en el Sistema.	Si existe un problema en la red u otro referente a la base de datos, no podrá desplegarse la pantalla correspondiente a la selección y se mostrará un mensaje de "ERROR, contacte a su administrador".	
2	No hay registro.	La información solicitada no está registrada, es decir, no se encontró ningún dato en la base de datos que coincida con la solicitud hecha. El Sistema mandará el siguiente mensaje: "No se encontraron registros".	
3	Información no disponible temporalmente.	No se mostrará la información solicitada en caso de que ésta esté siendo modificada en ese momento. El mensaje que se enviará al usuario será: "Información no disponible temporalmente. Intente más tarde por favor".	
4	Suspensión temporal.	No se podrá realizar ningún cambio si otro usuario está editando la información en ese momento. Por lo que se enviará el siguiente mensaje al usuario: "No procede. Intente más tarde por favor".	
5	Actualización falló.	En caso de que la actualización no pueda realizarse con éxito en la base de datos, el sistema lanzará el siguiente mensaje en la pantalla: "ERROR, la actualización no pudo ser realizada con éxito".	
6	Cancelación.	En caso de elegir "Cancelar", el sistema no procede con la petición anteriormente hecha.	
7	Eliminación falló.	Si el Sistema no puede llevar a cabo la eliminación de la información ya sea de forma parcial o total, se lanzará el siguiente mensaje "ERROR, no se ha podido completar la eliminación".	
8	Información incompleta.	Falta llenar algunos campos que son de carácter obligatorio. Aparecerá el siguiente mensaje "Campo obligatorio".	
9	Datos incongruentes.	Los datos introducidos no corresponden al tipo de dato asignado para ese campo. Se avisa al usuario mediante el siguiente mensaje: "Dato incorrecto" en el campo correspondiente.	

CASO DE USO		"ADMINISTRAR GRUPOS"
No. CU		<u>09</u>
EXCEPCIONES		
<i>E</i>	<i>Tipo</i>	<i>Acción</i>
10	Registro no completado.	Si existe un problema con la conexión u otro referente a la base de datos. No se podrá insertar el registro en ella, por lo que se lanzará un mensaje de "ERROR en la base de datos, no se pudo completar el registro".
11	Registro ya existente.	Si el sistema ya tiene registrada la información, se le notificará al usuario mediante el siguiente mensaje: "Esta (información) ya ha sido agregada".
POST-CONDICIONES		El usuario puede: <ul style="list-style-type: none"> • Administrar grupos (CU09). • Administrar usuarios (CU10) • Consultar registros (CU04) • Salir del sistema (CU03). • Ir a la página de inicio.

CASO DE USO		"ADMINISTRAR USUARIOS"	
No. CU		<u>10</u>	
ACTORES		Administrador.	
DESCRIPCIÓN		Permite al usuario administrar los usuarios del sistema mediante altas, bajas, cambios, consultas, activación/desactivación de cuentas y modificación de password.	
PRE-CONDICIONES		Del proceso: <ul style="list-style-type: none"> • El usuario ingresó al sistema de bases de datos de RedUNAM (CU_01). • El usuario fue autorizado (CU_02). 	
FLUJO PRINCIPAL			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
1	El usuario selecciona la opción del menú principal concerniente a los usuarios		
2	El sistema despliega todos los registros existentes correspondientes a los usuarios.		E1, E2
3	El usuario puede realizar: <ul style="list-style-type: none"> 3.1 Activar/desactivar usuarios. 3.2 Editar usuarios. 3.3 Borrar usuarios. 3.4 Agregar usuarios. 3.5 Cambiar password a los usuarios. 		
4	El sistema se actualiza y muestra todos los registros existentes correspondientes a los usuarios.		E5, E7
FLUJOS ALTERNOS			
"3.1 ACTIVAR/DESACTIVAR USUARIOS"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
3.1.1	El usuario elige uno de los registros desplegados por el sistema, seleccionando "Activo"		
3.1.2	El sistema despliega un mensaje que informa que se activará o desactivará al usuario seleccionado.		E1, E3
3.1.3	El sistema informa la activación/desactivación realizada. Ir a paso 4.		
"3.2 EDITAR USUARIOS"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
3.2.1	El usuario elige uno de los registros desplegados por el sistema, seleccionando "Editar".		
3.2.2	Se despliega la pantalla de edición en una nueva pestaña.		E1
3.2.3	El usuario realiza los cambios necesarios en los campos correspondientes y selecciona "Actualizar". Ir a paso 4.		E4, E6, E8, E9

CASO DE USO		"ADMINISTRAR USUARIOS"	
No. CU		<u>10</u>	
FLUJOS ALTERNOS			
"3.3 BORRAR USUARIOS"			
<i>Paso</i>	<i>Acción</i>		<i>Excepción</i>
3.3.1	El usuario elige uno de los registros desplegados por el sistema, seleccionando "Borrar".		
3.3.2	El Sistema despliega el siguiente mensaje: ¿Está seguro de borrar "el registro"? Aceptar /Cancelar".		E1
3.3.3	El usuario procede con la eliminación, es decir, elige "Aceptar".		E6
3.3.4	Se elimina la información de la base de datos. Ir a paso 4.		
"3.4 AGREGAR USUARIOS"			
3.4.1	El usuario selecciona agregar nuevo usuario.		
3.4.2	Se despliegan los campos a ser llenados.		E1
3.4.3	El usuario introduce los datos requeridos del nuevo usuario y selecciona agregar.		E6, E8, E9
3.4.4	El sistema envía la información a la base de datos y se almacenan los datos. Ir a paso 4		E10, E11
"3.5 CAMBIAR PASSWORD A LOS USUARIOS"			
3.5.1	El usuario elige uno de los registros desplegados por el sistema, seleccionando "Cambiar password".		
3.5.2	Se despliega la pantalla de edición de password en una nueva pestaña.		E1
3.5.3	El usuario proporciona la nueva contraseña y su respectiva verificación y selecciona "Guardar".		E6, E8, E9
3.5.4	El sistema envía la información a la base de datos y se almacenan los datos. Ir a paso 4.		E10, E11
EXCEPCIONES			
<i>E</i>	<i>Tipo</i>	<i>Acción</i>	
1	Problemas en el Sistema.	Si existe un problema en la red u otro referente a la base de datos, no podrá desplegarse la pantalla correspondiente a la selección y se mostrará un mensaje de "ERROR, contacte a su administrador".	
2	No hay registro.	La información solicitada no está registrada, es decir, no se encontró ningún dato en la base de datos que coincida con la solicitud hecha. El Sistema mandará el siguiente mensaje: "No se encontraron registros".	
3	Información no disponible temporalmente.	No se mostrará la información solicitada en caso de que ésta esté siendo modificada en ese momento. El mensaje que se enviará al usuario será: "Información no disponible temporalmente. Intente más tarde por favor".	
4	Suspensión temporal.	No se podrá realizar ningún cambio si otro usuario está editando la información en ese momento. Por lo que se enviará el siguiente mensaje al usuario: "No procede. Intente más tarde por favor".	
5	Actualización falló.	En caso de que la actualización no pueda realizarse con éxito en la base de datos, el sistema lanzará el siguiente mensaje en la pantalla: "ERROR, la actualización no pudo ser realizada con éxito".	
6	Cancelación.	En caso de elegir "Cancelar", el sistema no procede con la petición anteriormente hecha.	

CASO DE USO		"ADMINISTRAR USUARIOS"
No. CU		<u>10</u>
EXCEPCIONES		
<i>E</i>	<i>Tipo</i>	<i>Acción</i>
7	Eliminación falló.	Si el Sistema no puede llevar a cabo la eliminación de la información ya sea de forma parcial o total, se lanzará el siguiente mensaje "ERROR, no se ha podido completar la eliminación".
8	Información incompleta.	Falta llenar algunos campos que son de carácter obligatorio. Aparecerá el siguiente mensaje "Campo obligatorio".
9	Datos incongruentes.	Los datos introducidos no corresponden al tipo de dato asignado para ese campo. Se avisa al usuario mediante el siguiente mensaje: "Dato incorrecto" en el campo correspondiente.
10	Registro no completado.	Si existe un problema con la conexión u otro referente a la base de datos. No se podrá insertar el registro en ella, por lo que se lanzará un mensaje de "ERROR en la base de datos, no se pudo completar el registro".
11	Registro ya existente.	Si el sistema ya tiene registrada la información, se le notificará al usuario mediante el siguiente mensaje: "Esta (información) ya ha sido agregada".
POST-CONDICIONES		El usuario puede: <ul style="list-style-type: none"> • Administrar grupos (CU09). • Administrar usuarios (CU10) • Consultar registros (CU04) • Salir del sistema (CU03). • Ir a la página de inicio.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

PROPUESTA

Para poder abordar el problema descrito en el capítulo anterior, se propone realizar un sistema de bases de datos vía web donde se puedan realizar las acciones que se requieren tales como: altas, bajas, cambios y consultas básicamente, para lo que se tomará como guía el diagrama general de casos de uso (Figura 2.6) y las especificaciones correspondientes a cada uno de ellos.

El diseño del sistema estará formado de manera principal por los siguientes componentes:

- Menú general, el cual estará disponible siempre y dará acceso a opciones importantes que se definirán más adelante.
- Pestañas, que simulen carpetas que se van abriendo de acuerdo a la selección correspondiente por parte del usuario, formando así un árbol jerárquico con lo que se pretende una mejor visualización y distribución de la información de la base de datos y mostrar la ubicación del usuario en cualquier momento, así como poder ofrecer una mejor accesibilidad y facilidad a cualquier acción que se requiera ejecutar.

Para llevarlo a cabo, se debe elegir una metodología a seguir y las herramientas adecuadas por lo que a continuación se realizan la elección de ellas y su respectiva justificación.

3.1 EVALUACIÓN Y SELECCIÓN DE HERRAMIENTAS

En el desarrollo de una aplicación de software se utilizan distintas herramientas, tales como lenguajes de programación, o bien, estructurados, sistemas de gestión de bases de datos, servidores, lenguajes de marcas, sistemas operativos, entre otras.

Existe una gran cantidad y diversidad de dichas herramientas, algunas son software libre y otras software propietario. Cada una de ellas poseen características específicas que ofrecen ciertas ventajas, mismas que nos brindan grandes oportunidades de desarrollo pero a su vez también presentan algunas desventajas, pudiendo llegar a la conclusión de que un aspecto determinante en la elección de ellas, es el proyecto que se quiere desarrollar ya que se tienen que contemplar los requerimientos, el presupuesto, el equipo, entre otros factores. Asimismo, los conocimientos y la experiencia del desarrollador del sistema y el tiempo en el que se requiere tenerlo ya terminado y en funcionamiento.

Por lo tanto, para elegir las herramientas que se utilizarán en el desarrollo de este sistema se han considerado los siguientes factores:

- 1) El cliente ha sugerido el uso de software libre.
- 2) La propuesta presentada para el proyecto.
- 3) Experiencia propia en el uso de las herramientas.
- 4) Conocimiento de las herramientas.
- 5) Fácil manejo y soporte de las herramientas.
- 6) Recursos materiales disponibles.
- 7) Accesibilidad y disponibilidad de las herramientas.

Dichos factores han determinado que las herramientas seleccionadas para trabajar en este proyecto son:

- MySQL (Descrito en 1.1.4) como manejador de bases de datos.

Debido a que en el mundo del software libre de los manejadores de bases de datos se tienen dos opciones viables que ya se han mencionado previamente, las cuales son: MySQL y PostgreSQL. De las que se ha elegido MySQL debido a las ventajas que ofrece sobre PostgreSQL específicamente para este proyecto en la cuestión de la compatibilidad y la integridad con PHP pero también porque es un DBMS con el cual ya he interactuado en ocasiones anteriores.

- Apache (Descrito en 1.1.6) como servidor web.

Por ser el servidor web libre mayor difundido y con los mejores resultados a nivel mundial y personal, así como su fácil configuración.

- PHP (Descrito en 1.4.5) como lenguaje de programación interpretado para la creación del sitio web y por consiguiente HTML (Descrito en 1.4.1) como lenguaje de marcas, uso de JavaScript (Descrito en 1.4.3) y Ajax (Descrito en 1.4.4).
Porque es un lenguaje de programación muy robusto con sintaxis basada en C, Java y Perl, lo cual facilitará la comprensión del mismo y el desarrollo del proyecto dado mi experiencia en su uso. Además de que tiene soporte para PostgreSQL y posee gran funcionalidad. Cabe mencionar que PHP estará embebido en páginas HTML donde se podrá enriquecer el código con el uso de Javascript y Ajax, todo esto en conjunto logra un excelente funcionamiento, que brinda muchas ventajas porque aumenta la interactividad, velocidad y usabilidad en la aplicación, es por esta razón que también se decidió incorporarlos.

- OpenBSD (UNIX) como sistema operativo del servidor.

Ya que el software utilizado es compatible con esta plataforma y debido a la facilidad de mantenimiento y la seguridad en él. Así como el respaldo dado por la experiencia en el uso de este sistema en otros servidores dentro de NIC-UNAM.

Respecto al software libre, cabe resaltar que debido a su amplia distribución, está perfectamente soportado por una gran comunidad de desarrolladores, lo cual será un gran apoyo ante cualquier obstáculo que se presente en cualquier etapa del proceso de desarrollo del software.

3.2 METODOLOGÍA

Las metodologías existentes nos permiten llevar a cabo el desarrollo de un software, depende de los requerimientos del proyecto a realizar así como de las necesidades a satisfacer, la elección de una de ellas ya que poseen características, fases, principios, etc., propios, que marcan la diferencia entre una y otra, tal como se presenta en la tabla 1.16, donde podemos observar las características principales así como sus ventajas y desventajas de las metodologías más conocidas y aplicadas en este campo.

De acuerdo a lo anterior, se realizó un análisis exhaustivo de dichas metodologías y con base en el análisis del sistema descrito previamente en el capítulo 2, se ha determinado que la metodología que se usará en el desarrollo de esta tesis será la de Rational Unified Process (RUP), la cual está basada en UML y se caracteriza por ser:

- **Proceso Dirigido por los Casos de Uso.-** Con esto se refiere a la utilización de los casos de uso para el desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Los casos de uso son la base para la implementación de las fases y disciplinas del RUP. Un caso de uso es una secuencia de pasos a seguir para la realización de un fin o propósito, y se relaciona directamente con los requerimientos, ya que un caso de uso es la secuencia de pasos que conlleva la realización e implementación de un requerimiento planteado por el cliente.
- **Proceso Iterativo e Incremental.-** Es el modelo utilizado por RUP para el desarrollo de un proyecto de software. Este modelo plantea la implementación del proyecto a realizar en iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración, con lo cual se tienen varias ventajas, entre ellas se puede mencionar la de tener pequeños avances del proyectos que son entregables al cliente el cual puede probar mientras se está desarrollando otra iteración del proyecto, con lo cual el proyecto va creciendo hasta completarlo en su totalidad.

- **Proceso Centrado en la Arquitectura.-** Define la arquitectura de un sistema, y una arquitectura ejecutable construida como un prototipo evolutivo. Arquitectura de un sistema es la organización o estructura de sus partes más relevantes. Una arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades. RUP establece refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo evolutivo.

Esta metodología proporciona todas las bases y principios para llevar a cabo la elaboración del software con éxito. Los cuales se presentan a continuación:

1. Adaptar el proceso.
2. Balancear prioridades
3. Colaboración entre equipos.
4. Demostrar valor iterativamente.
5. Elevar el nivel de abstracción.
6. Enfocarse en la calidad.

Lo cual involucra lo siguiente:

- Mitigar los errores y riesgos de manera temprana y continua.
- Asegurarse del valor y calidad del producto que se le entregará al cliente.
- Permanecer atentos en el software ejecutable que se va obteniendo.
- Incorporar los cambios tempranamente en el proyecto.
- Construir el sistema por medio de componentes.
- Trabajar todos como un solo equipo.
- Establecer una arquitectura ejecutable al iniciar el proyecto.
- Adoptar la calidad como una forma de vida y no como un cambio.

RUP es considerado un conjunto de metodologías adaptables al contexto y necesidades de cada organización y no un sistema con pasos firmemente establecidos, por lo que brinda flexibilidad. Consta de dos dimensiones:

1. El eje horizontal representa el tiempo y muestra los aspectos del ciclo de vida del proceso. Se refiere al aspecto dinámico del proceso conforme se va desarrollando y se expresa en términos de fases, de iteraciones, y la finalización de las fases.
2. El eje vertical representa las disciplinas, que agrupan actividades de manera lógica de acuerdo a su naturaleza. Se refiere al aspecto estático del proceso: cómo se describe en términos de componentes del proceso, las disciplinas, las actividades, los flujos de trabajo, los artefactos y los roles.

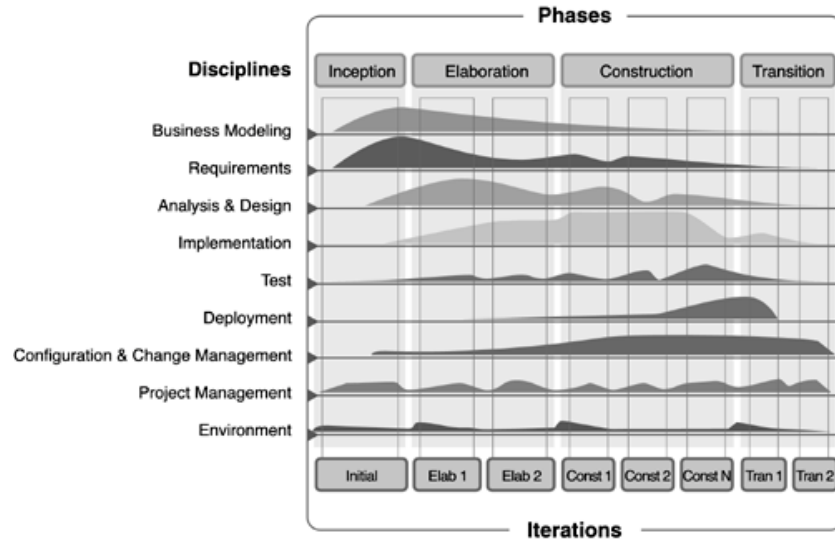


Figura 3.1 Las dos dimensiones de RUP (Kroll, 2003).

Donde las fases se refieren a lo siguiente:

- **Inicio o concepción:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. El objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos. En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario. Básicamente se debe obtener la capacidad operacional inicial.
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

Como vemos, RUP se basa en componentes, lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas.

En la etapa de programación de la aplicación, se seguirá el patrón de diseño Modelo-Vista-Controlador, mismo que se describe a continuación:

MODEL-VIEW-CONTROLLER (MVC)

El Modelo Vista Controlador (MVC) es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres

componentes distintos. Es usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas de una manera paralela e independiente. Por lo que, el patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Bases de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Descripción del patrón

- **Modelo:** Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de éstos y permite derivar nuevos datos. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario o a la aplicación misma.
- **Vista:** Es la representación del modelo en forma gráfica mediante un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este componente responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Muchos sistemas informáticos utilizan un Sistema de Gestión de Bases de Datos para gestionar los datos: en MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio de capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario. Los controladores complejos

están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Cabe mencionar que en algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Ventajas y desventajas de MVC

Las principales ventajas de hacer uso del patrón MVC son:

- La separación del Modelo de la Vista, es decir, separar los datos de la representación visual de los mismos.
- Es mucho más sencillo agregar múltiples representaciones de los mismos datos e información.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.
- Permite el escalamiento de la aplicación en caso de ser requerido.

Las desventajas de seguir el planteamiento de MVC son:

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- La curva de aprendizaje del patrón de diseño es más alta que usando otros modelos más sencillos.

¿Por qué utilizar MVC?

Porque es un patrón de diseño de software probado y la aplicación se puede desarrollar rápidamente, de forma modular y mantenible. Separar las funciones de la aplicación en modelos, vistas y controladores hace que la aplicación sea muy ligera. Las características nuevas se añaden fácilmente y las antiguas toman automáticamente una forma nueva.

El diseño modular permite a los diseñadores y a los desarrolladores trabajar conjuntamente, así como realizar rápidamente el prototipado. Esta separación también permite hacer cambios en una parte de la aplicación sin que las demás se vean afectadas.

Actualmente existen muchos frameworks disponibles para el desarrollo de aplicaciones MVC en distintos lenguajes, algunos de ellos se muestran en la siguiente tabla:

Lenguaje	Licencia	Nombre
Ruby	MIT	Ruby on Rails
Java / J2ee	Apache	Struts
Java / J2ee	Apache	Beehive
Java / J2ee	Apache	Spring
Java / J2ee	Apache	Tapestry
Java / J2ee	Apache	Aurora
Java / J2ee	Apache	JavaServerFaces
Perl	GPL	Catalyst
Perl	GPL	CGI::Application
Perl	GPL	Gantry Framework
Perl	GPL	Jifty
Perl	GPL	Maypole
Perl	GPL	OpenInteract2
Perl	Comercial	PageKit
Perl	GPL	Cyclone 3
Perl	ECL	Solstice
Perl	GPL	CGI::Builder
PHP	GPL	Self Framework
PHP	LGPL	Tlalokes
PHP	LGPL	Agavi
PHP	BSD	Zend Framework
PHP	MIT	CakePHP
PHP	GNU/GPL	Kumbia
PHP	MIT	Symfony
PHP	MIT	QCodo
PHP	GNU/GPL	CodeIgniter
PHP	Otra	Kohana
PHP	MPL 1.1	PHP4ECore
PHP	BSD	PRADO
Python	ZPL	Zope3
Python	Varias	Turbogears
Python	GPL	Web2py
Python	BSD	Django
.NET	Castle Project	MonoRail
.NET	Apache	Spring .NET
.NET	Apache	Maverick .NET
.NET	MS-PL	ASP.NET MVC
.NET	Microsoft Patterns & Practices	User Interface Process (UIP) Application Block
AS3	Adobe Open Source	Cairngorm

Tabla 3.1 Frameworks MVC.

El framework que se utilizará será CakePHP, debido a que se definió a PHP como lenguaje de programación y porque brinda las bases adecuadas para lo que se quiere realizar, como podemos observar a continuación detalladamente:

CakePHP

CakePHP es un framework de desarrollo de aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails, libre, de código abierto, que implementa el patrón MVC.

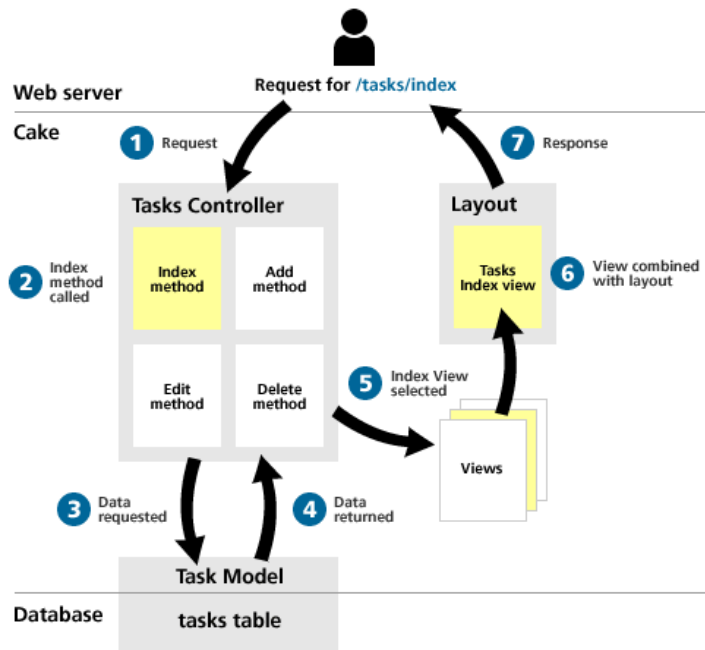


Figura 3.2 Funcionamiento básico de CakePHP

- 1) El usuario da click sobre la liga `http://server.org/tasks/index`.
- 2) El dispatcher (despachador) de CakePHP llama y ejecuta el controlador `tasks_controller.php` el cual (como se indica en el URL) tiene un método `index()`.
- 3) Este método `index()` del controlador llama al modelo `Task` que hace una consulta a la base de datos SQL.
- 4) El modelo regresa el resultado al controlador.
- 5) El controlador ha concluido su trabajo y le pasa los datos a la vista.
- 6) La vista se integra con el layout HTML.
- 7) El usuario ve el resultado de pinchar sobre `http://server.org/tasks/index`.

Características

- Licencia flexible
- Compatible con PHP4 y PHP5
- CRUD integrado para la interacción con la base de datos
- Soporte de aplicación [scaffolding]
- Generación de código
- Arquitectura Modelo Vista Controlador (MVC)
- Despachador de peticiones [dispatcher], con URLs y rutas personalizadas y limpias
- Validación integrada
- Plantillas rápidas y flexibles (sintaxis de PHP, con ayudantes[helpers])
- Ayudantes para AJAX, Javascript, formularios HTML y más
- Componentes de Email, Cookie, Seguridad, Sesión y Manejo de solicitudes
- Listas de control de acceso flexibles
- Limpieza de datos
- Caché flexible
- Localización
- Funciona en cualquier subdirectorio del sitio web, con poca o ninguna configuración de Apache
- Comunidad activa y amistosa

¿Por qué utilizar CakePHP?

Además de las características mencionadas anteriormente, también porque este framework proporciona una estructura de organización básica, desde los nombres de los archivos hasta los de las tablas de la base de datos, manteniendo toda la aplicación consistente y lógica.

CakePHP incluye las clases Controlador [Controller], Modelo [Model] y Vista [View], pero también incluye otras clases y objetos que hacen que el desarrollo en MVC sea un poco más rápido y agradable. Los Componentes [Components], Comportamientos [Behaviors], y Ayudantes [Helpers] son clases que proporcionan extensibilidad y reusabilidad; agregan rápidamente funcionalidad a las clases base MVC de las aplicaciones.

CAPÍTULO 4

DESARROLLO DEL SISTEMA

4.1 DISEÑO Y CREACIÓN DEL SISTEMA

Las bases para el diseño del sistema fueron establecidas tomando en cuenta la propuesta expuesta en el capítulo 3, así como los siguientes aspectos:

- 1) Antecedentes (La base de datos anterior).
- 2) Incorporación de la red de voz.
- 3) Implementación de nuevos módulos.
- 4) Reestructuración de la base de datos.
- 5) Peticiones de los usuarios.
- 6) Experiencia personal de la interacción con la base de datos.

Para comprender mejor los casos de uso presentados en el capítulo 2, a continuación se muestran los diagramas de actividades del sistema, los cuales representan el comportamiento interno de los casos de uso, bajo la forma de un desarrollo por etapas, agrupadas secuencialmente, básicamente son la serie de actividades que deben ser realizadas en un caso de uso. Estos diagramas se tomaron como guías para el desarrollo del sistema:

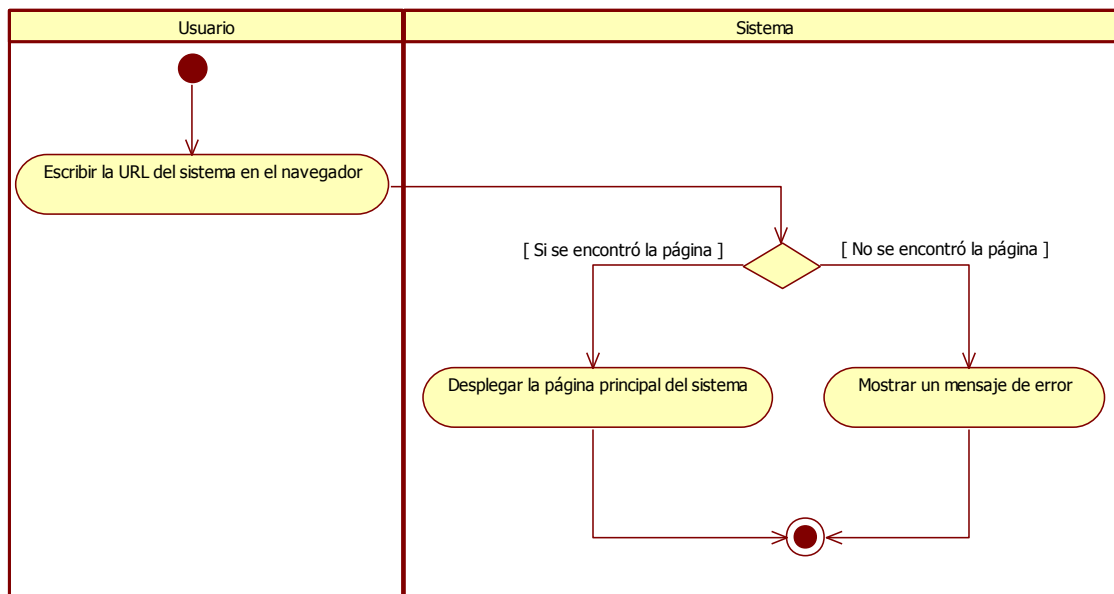


Figura 4.1 Diagrama de actividad "Entrar al sistema".

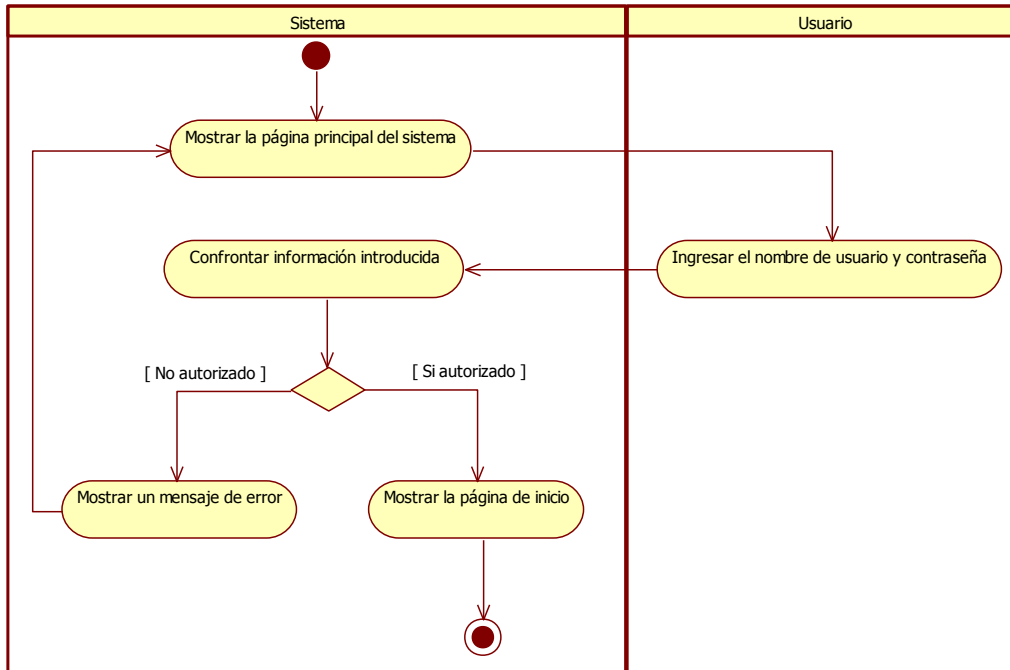


Figura 4.2 Diagrama de actividad “Autenticar los datos del usuario en el sistema”.

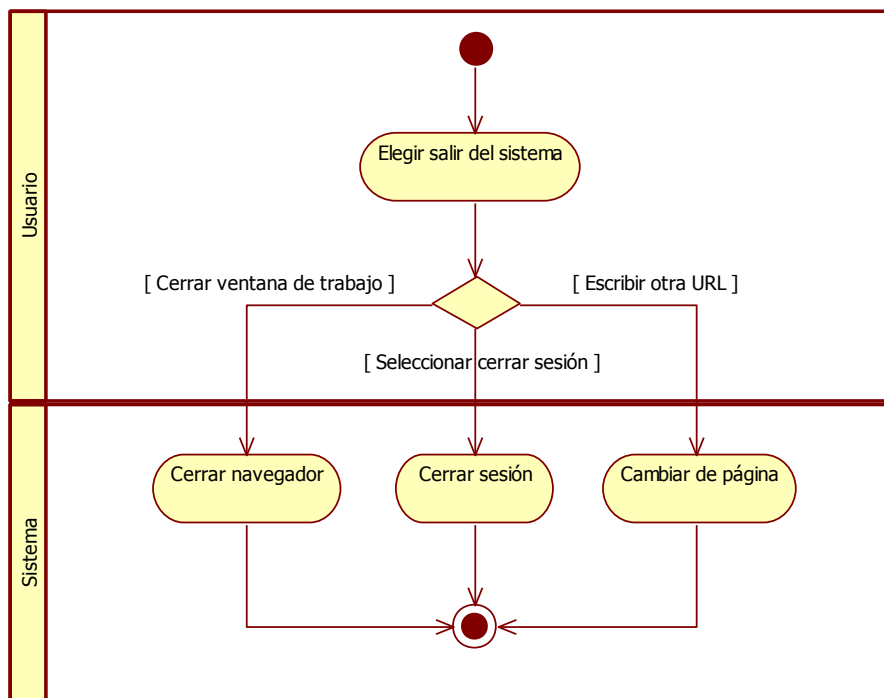


Figura 4.3 Diagrama de actividad “Salir del sistema”.

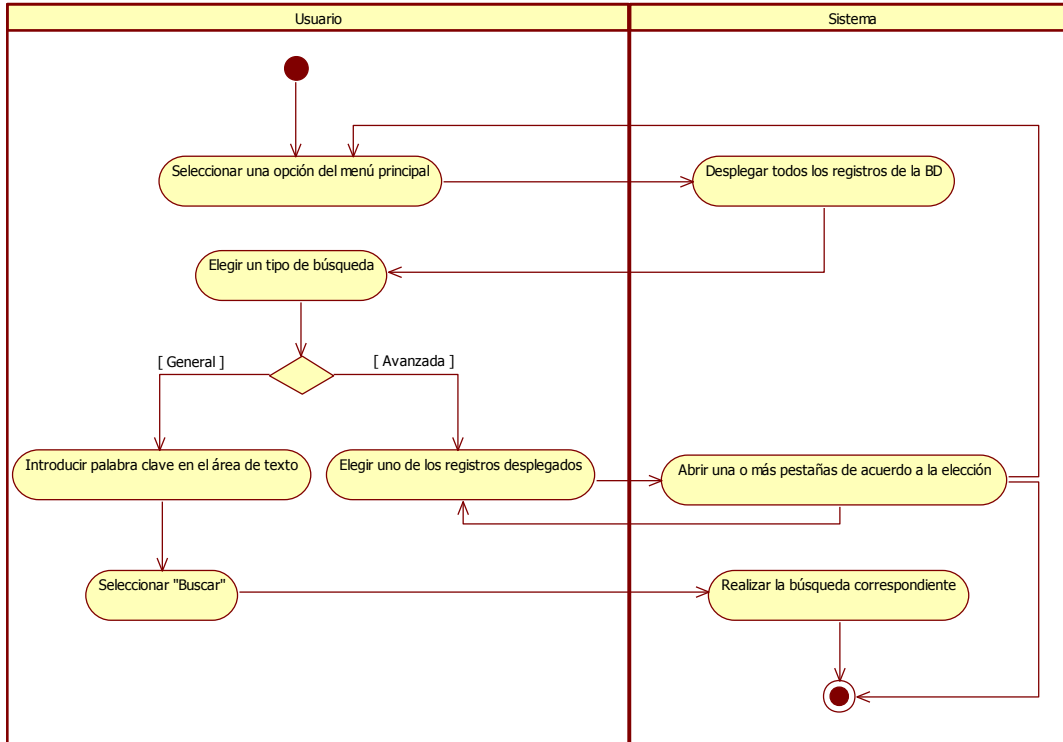


Figura 4.4 Diagrama de actividad “Consultar registros mediante búsqueda en la base de datos”.

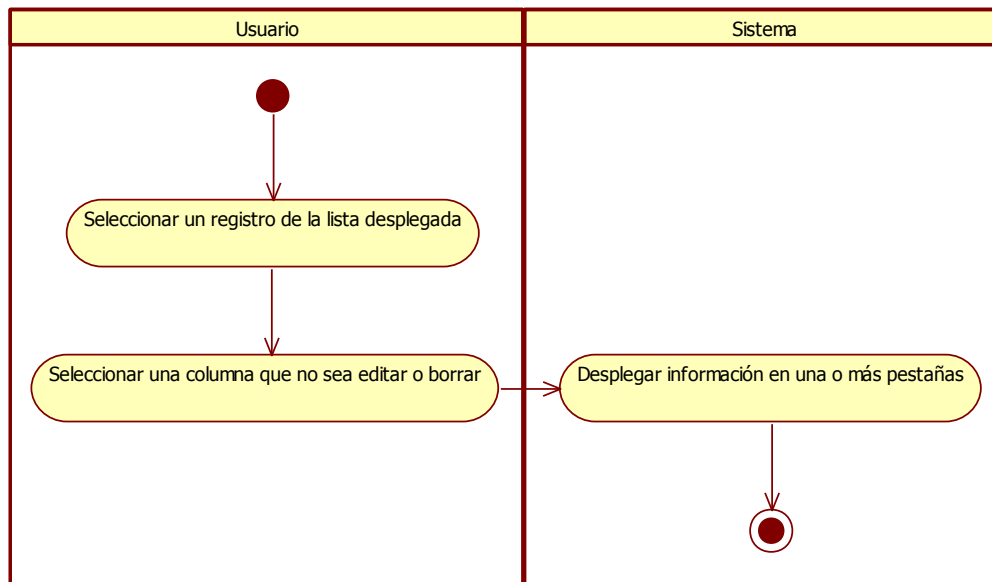


Figura 4.5 Diagrama de actividad “Consultar información acerca de un registro en específico”.

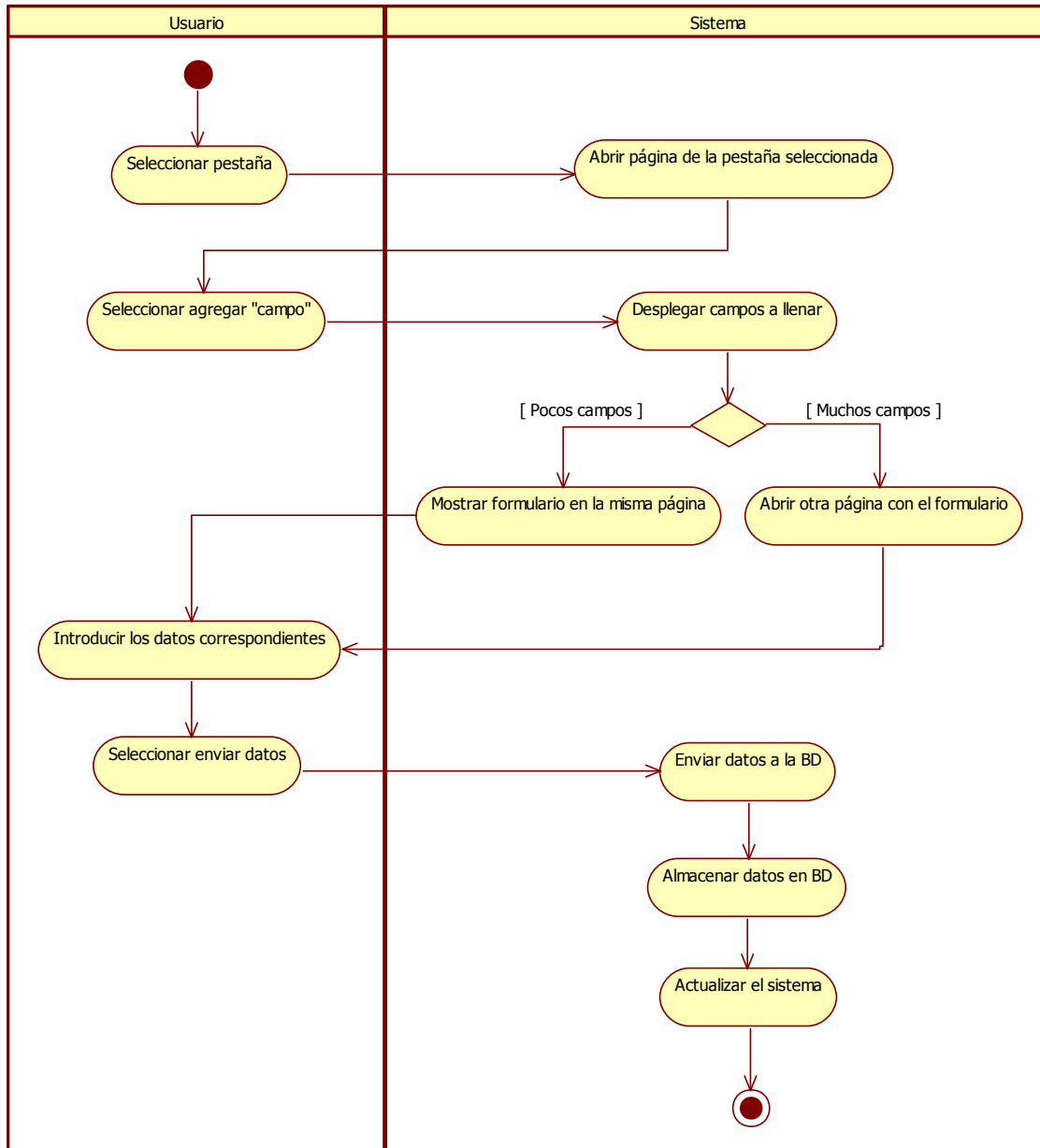


Figura 4.6 Diagrama de actividad “Agregar un registro a la base de datos”.

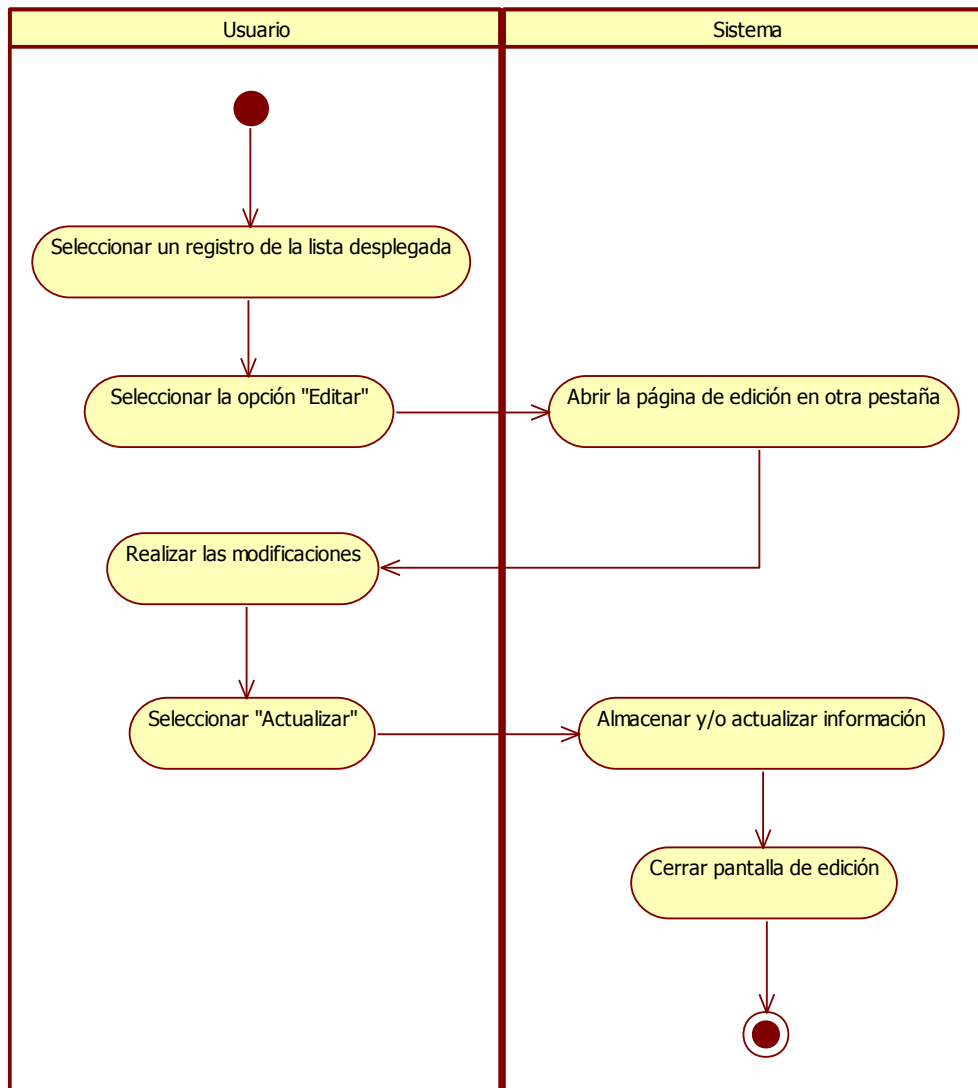


Figura 4.7 Diagrama de actividad “Modificar un registro de la base de datos”.

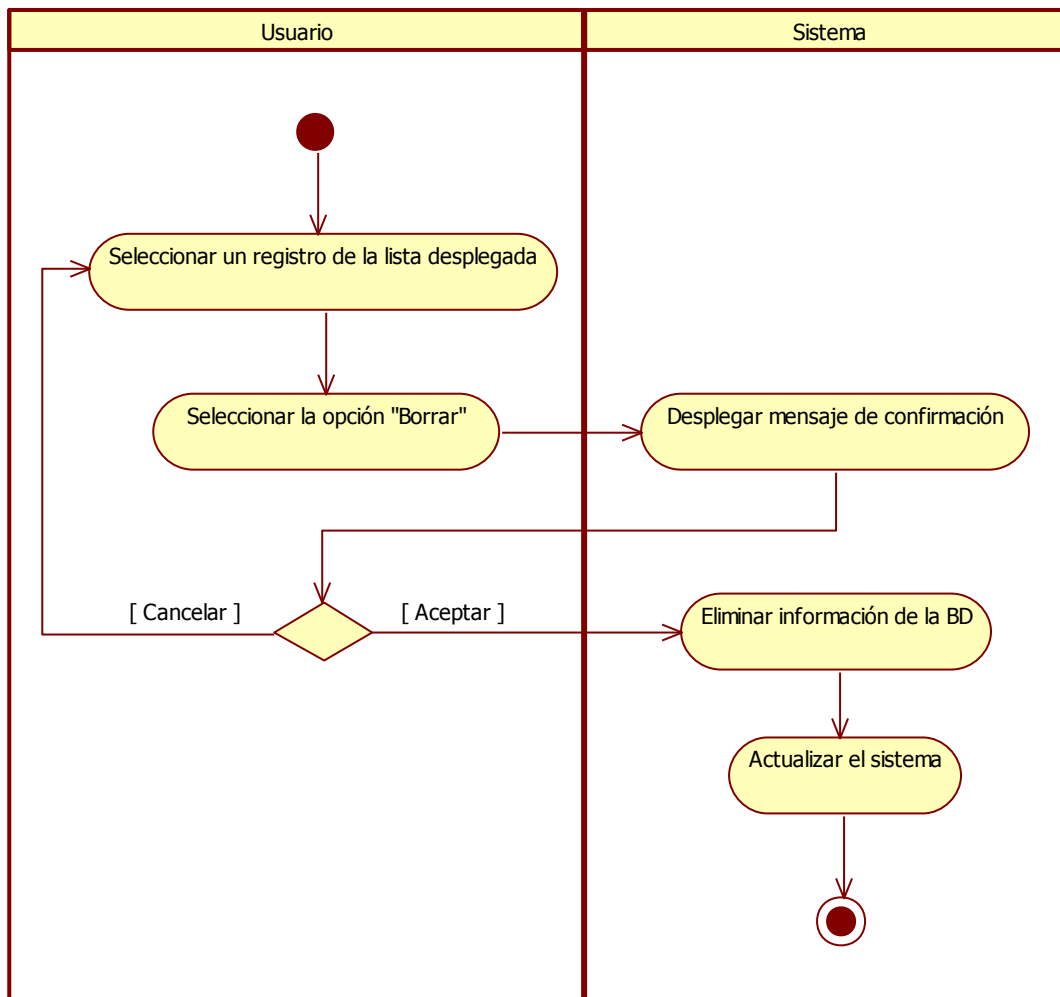


Figura 4.8 Diagrama de actividad “Eliminar un registro de la base de datos”.

A continuación se presenta el diagrama de navegación utilizando un diagrama de estados donde se muestra el flujo de interacción dentro de la interfaz de usuario:

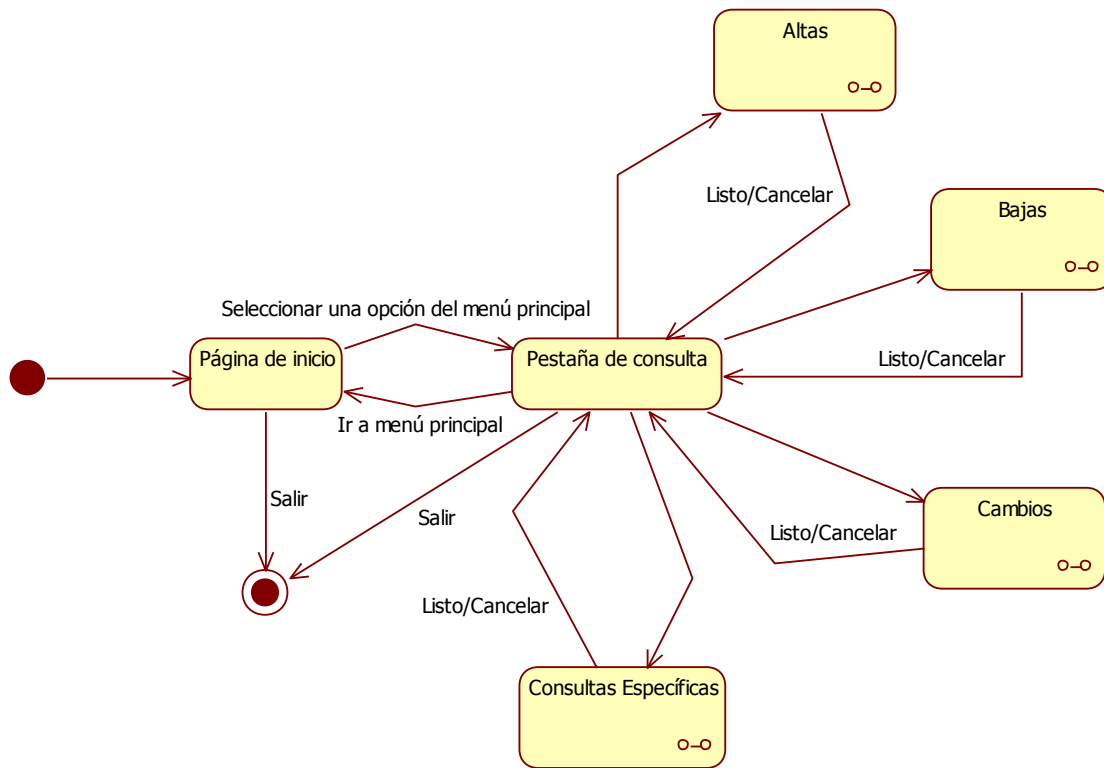


Figura 4.9 Diagrama de navegación del sistema.

Para abordar el hardware del sistema, se hará uso del diagrama de distribución, el cual se enfoca primordialmente a esta parte ya que este tipo de diagramas muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.

Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene algo de memoria y, a menudo, capacidad de procesamiento.

Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes.

A continuación se muestra el diagrama del Sistema de Bases de Datos para las Redes de Voz y Datos de la UNAM:

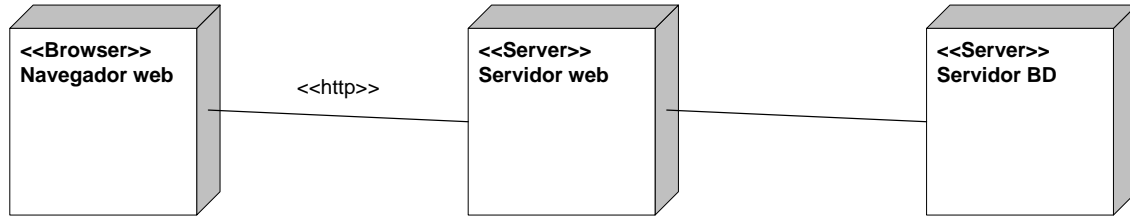


Figura 4.10 Diagrama de distribución del sistema.

Se implementará la arquitectura cliente-servidor de tres capas (descrita en 1.1.1) donde se disponen de tres tipos de nodos:

- Clientes que interactúan con los usuarios finales.
- Servidores de aplicación que procesan los datos para los clientes.
- Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

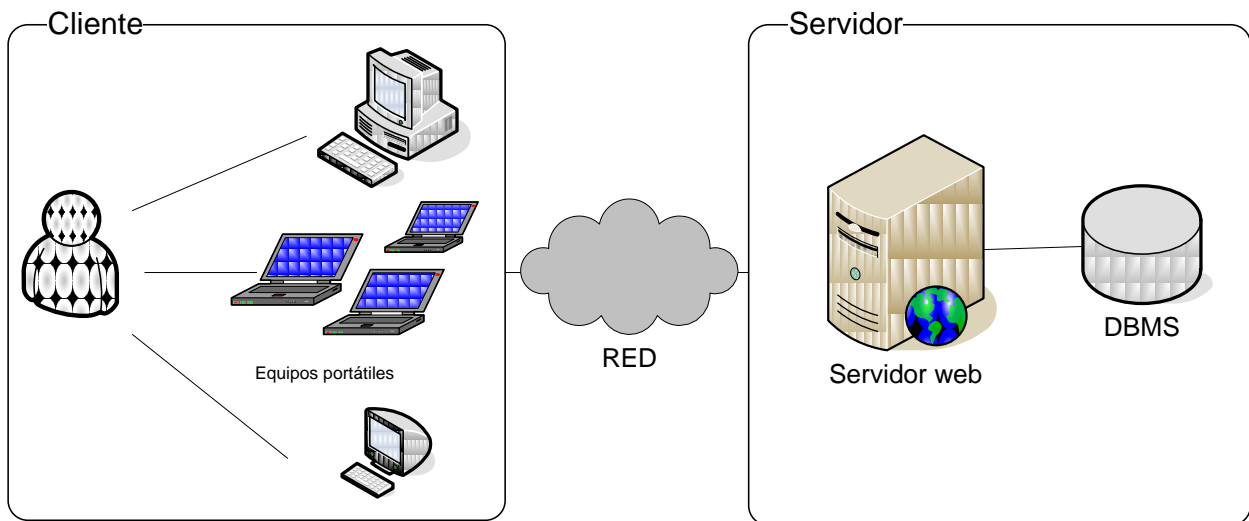


Figura 4.11 Arquitectura del sistema (cliente-servidor de tres capas).

El sistema se encuentra soportado por una base de datos, la cual se planteó bajo el modelo relacional (RDBMS), por lo que a continuación se presenta la estructura lógica general de la base de datos, es decir el modelo entidad relación, el cual es necesario para poder obtener las tablas correspondientes al modelo relacional:

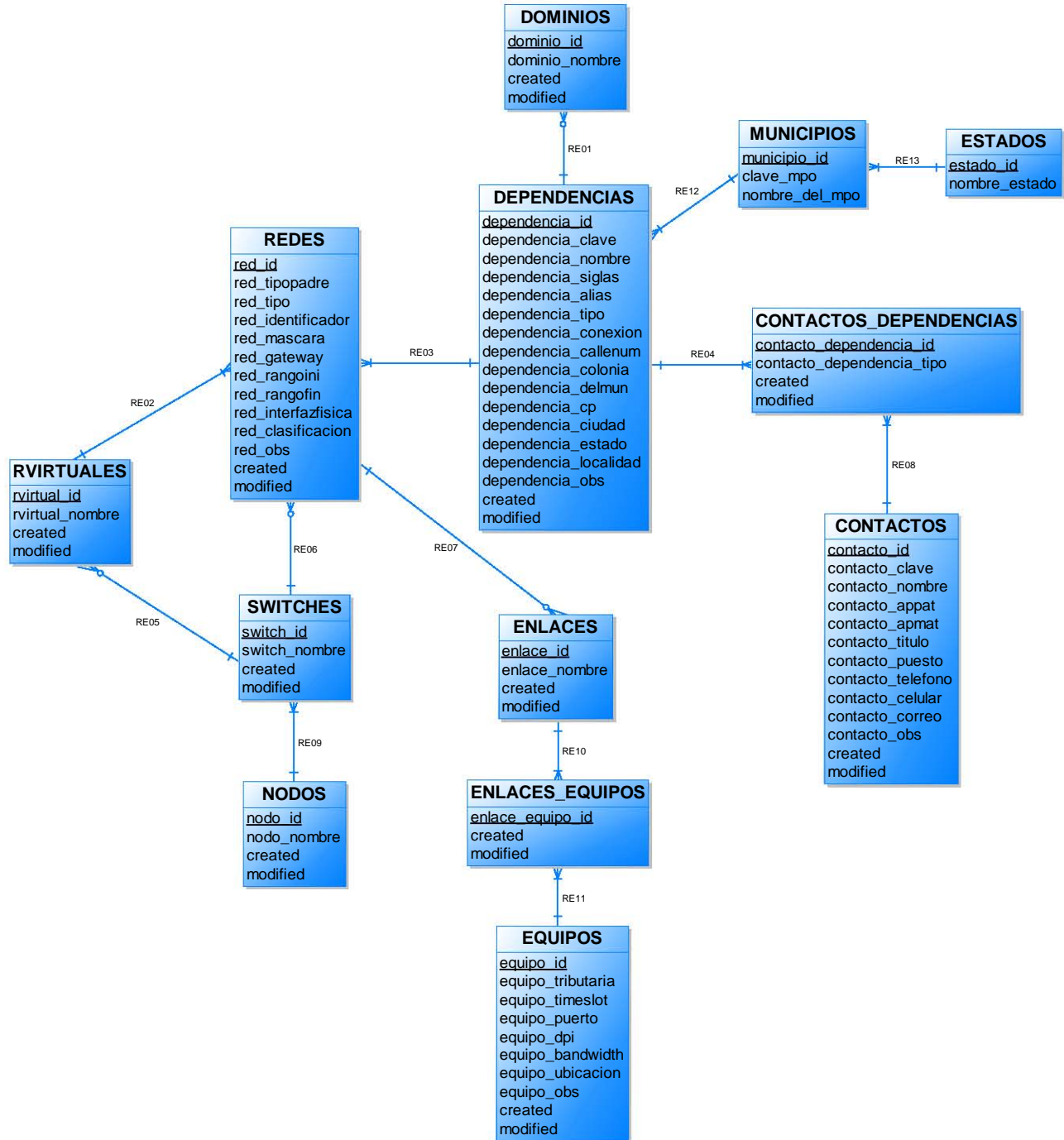


Figura 4.12 Modelo Entidad-Relación Base.

Con base en el modelo E-R Base (Figura 4.12) se generan las tablas que integrarán la base de datos, quedando de la siguiente manera:



Figura 4.13 Modelo Relacional Base.

A continuación se presentan los modelos de configuración del sistema:

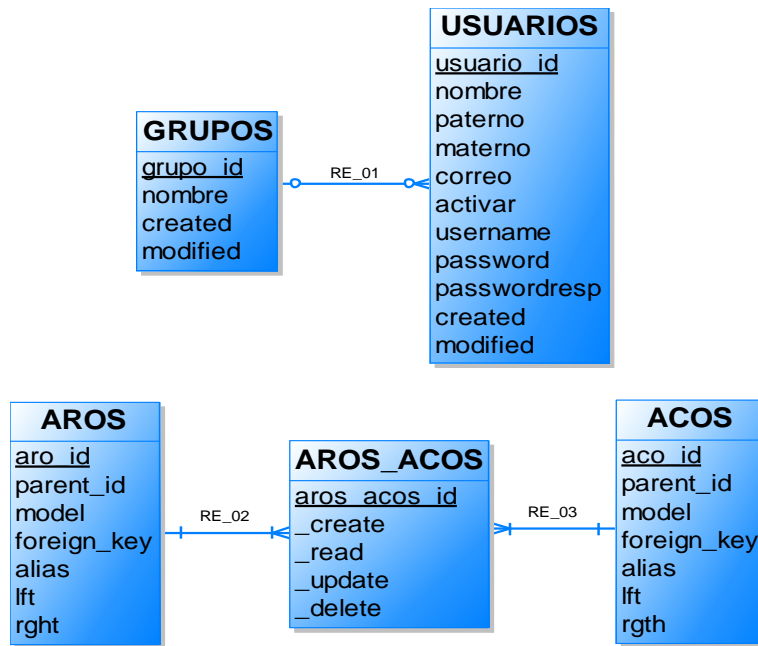


Figura 4.14 Modelo Entidad-Relación de Configuración.

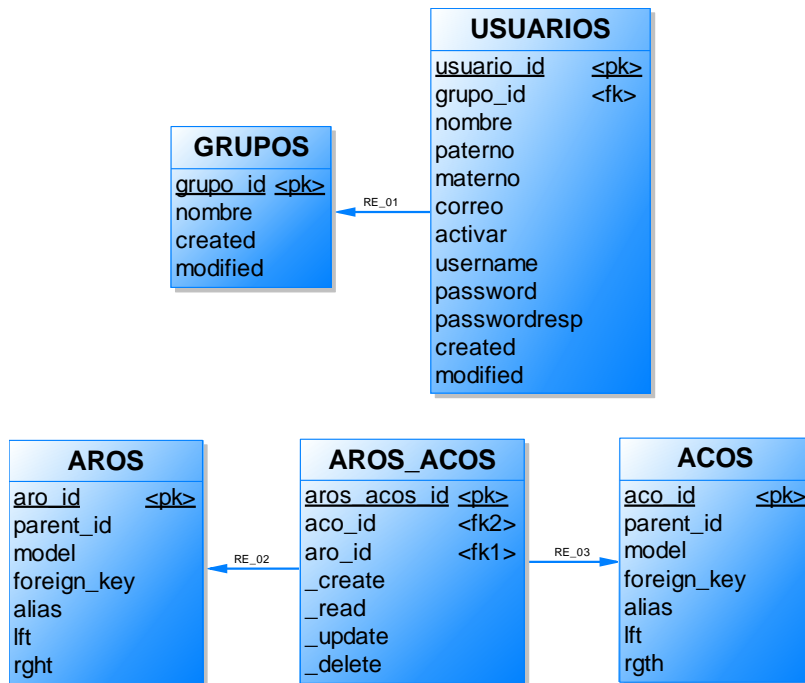


Figura 4.15 Modelo Relacional de Configuración.

También es presentado el respectivo diccionario de datos, donde podemos encontrar cada detalle importante referente a los atributos de las tablas, con el fin de tener un pleno control de los datos.

DICCIONARIO DE DATOS

TABLA NODOS.- Tiene relación con la tabla Switches.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave Primaria	Llave externa
nodo_id	Identificador de Nodo	int		Not null	✓	
nodo_nombre	Nombre del nodo	varchar	15	Not null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA SWITCHES.- Tiene relación con las tablas: Nodos, Rvirtuales y Redes.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
switch_id	Identificador de Switch	int		Not null	✓	
nodo_id	Identificador perteneciente a Nodos	int		Not null		✓
switch_nombre	Nombre del switch	varchar	250	Not null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA RVIRTUALES.- Tiene relación con la tabla Switches y Redes.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
rvirtual_id	Identificador de Rvirtual	int		Not null	✓	
switch_id	Identificador perteneciente a Switches	int		Not null		✓
rvirtual_nombre	Nombre de la red virtual	varchar	150	Not null		
Created	Fecha de creación	date		Null		
Modified	Última modificación	date		Null		

TABLA DEPENDENCIAS.- Tiene relación con las tablas: Redes y Contactos_Dependencias.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
dependencia_id	Identificador de la dependencia	int		Not null	✓	
county_id	Identificador perteneciente a Counties	int		Not null		✓
dependencia_clave	Clave de la dependencia	varchar	12	Not null		
dependencia_nombre	Nombre de la dependencia	varchar	250	Not null		
dependencia_siglas	Siglas de la dependencia	varchar	20	Not null		
dependencia_alias	Alias de la dependencia	varchar	100	Null		
dependencia_tipo	Tipo de dependencia	boolean		Not null		
dependencia_conexion	Tipo de conexión dependencia	boolean		Not null		
dependencia_callenum	Calle y número de la dependencia	varchar	100	Null		
dependencia_colonia	Colonia de la dependencia	varchar	100	Null		
dependencia_delmun	Delegación y municipio de la dependencia	varchar	100	Null		
dependencia_cp	Código postal de la dependencia	int		Null		
dependencia_ciudad	Ciudad de la dependencia	varchar	100	Null		
dependencia_estado	Estado de la dependencia	varchar	100	Not null		
dependencia_localidad	Localidad de la dependencia	varchar	100	Null		
dependencia_obs	Observaciones de la dependencia	varchar	300	Null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA CONTACTOS.- Tiene relación con la tabla Contactos_Dependencias.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
contacto_id	Identificador del contacto	int			✓	
contacto_clave	Clave del contacto	varchar	10	Not null		
contacto_nombre	Nombre(s) del contacto	varchar	100	Not null		
contacto_appat	Apellido paterno del contacto	varchar	100	Not null		
contacto_apmat	Apellido materno del contacto	varchar	100	Null		
contacto_titulo	Profesión del contacto	varchar	100	Null		
contacto_puesto	Puesto del contacto	varchar	100	Null		
contacto_telefono	Teléfonos del contacto	varchar	100	Not null		
contacto_celular	Celular del contacto	varchar	30	Null		
contacto_correo	Correo electrónico del contacto	varchar	100	Not null		
contacto_obs	Observaciones del contacto	varchar	300	Null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA CONTACTOS DEPENDENCIAS.- Tiene relación con las tablas: Dependencias y Contactos.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
contacto_dependencia_id	Identificador del Contacto_Dependencia	int		Not null	✓	
contacto_id	Identificador perteneciente a Contactos	int		Not null		✓
dependencia_id	Identificador perteneciente a Dependencias	int		Not null		✓
contacto_dependencia_tipo	Tipo de contacto	boolean		Not null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA DOMINIOS.- Tiene relación con la tabla Dependencias.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
dominio_id	Identificador del dominio	int		Not null	✓	
dependencia_id	Identificador perteneciente a Dependencias	int		Not null		✓
dominio_nombre	Nombre del dominio	varchar	100	Not null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA REDES.- Tiene relación con las tablas: RVirtuales, Switches y Dependencias.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
red_id	Identificador de la Red	int		Not null	✓	
switch_id	Identificador perteneciente a Switches	int		Null		✓
rvirtual_id	Identificador perteneciente a Rvirtuales	int		Null		✓
dependencia_id	Identificador perteneciente a Dependencias	int		Not null		✓
red_tipopadre	Tabla padre	boolean		Not null		
red_tipo	Tipo de red	boolean		Not null		
red_identificador	Identificador de Red	varchar	16	Not null		
red_mascara	Máscara de la Red	varchar	16	Not null		
red_gateway	Gateway de la Red	varchar	16	Not null		
red_rangoini	Inicio del rango de la Red	int		Not null		
red_rangofin	Fin del rango de la Red	int		Not null		
red_interfazfisica	Interfaz física de la Red	varchar	100	Not null		
red_clasificacion	Clasificación de la Red	boolean		Not null		
red_obs	Observaciones de la Red	varchar	300	Null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA ENLACES.- Tiene relación con las tablas: Redes y Enlaces_Equipos.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
enlace_id	Identificador de Enlace	int		Not null	✓	
red_id	Identificador perteneciente a Redes	int		Not null		✓
enlace_nombre	Nombre del enlace	varchar	100	Not null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA EQUIPOS.- Tiene relación con la tabla Enlaces_Equipos.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
equipo_id	Identificador del Equipo	int		Not null	✓	
equipo_tributaria	Tributaria del Equipo	int		Not null		
equipo_timeslot	Time slot del Equipo	varchar	20	Not null		
equipo_puerto	Puerto del equipo	varchar	20	Not null		
equipo_dpi	DPI del equipo	varchar	40	Not null		
equipo_bandwidth	Ancho de banda del Equipo	int		Not null		
equipo_ubicacion	Ubicación del Equipo	varchar	100	Not null		
equipo_obs	Observaciones del Equipo	varchar	300	Null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA ENLACES EQUIPOS.- Tiene relación con las tablas: Enlaces y Equipos.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave Primaria	Llave externa
enlace_equipo_id	Identificador de Enlace_Equipo	int		Not null	✓	
enlace_id	Identificador perteneciente a Enlaces	int		Not null		✓
equipo_id	Identificador perteneciente a Equipos	Int		Not null		✓
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA MUNICIPIOS.- Tiene relación con las tablas: Dependencias y Estados.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave Primaria	Llave externa
municipio_id	Identificador de County	int		Not null	✓	
estado_id	Identificador perteneciente a States	int		Not null		✓
clave_mpo	Clave del municipio	varchar	3	Not null		
nombre_del_mpo	Nombre del municipio	varchar	200	Not null		

TABLA ESTADOS.- Tiene relación con las tabla Municipios.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave Primaria	Llave externa
estado_id	Identificador de County	int		Not null	✓	
nombre_estado	Nombre del Estado	varchar	200	Not null		

TABLA GRUPOS.- Tiene relación con las tablas Usuarios.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave Primaria	Llave externa
grupo_id	Identificador de Grupo	int		Not null	✓	
nombre	Nombre del grupo del sistema	varchar	50	Not null		
created	Fecha y hora de creación	datetime		Not null		
modified	Última modificación	datetime		Not null		

TABLA USUARIOS.- Tiene relación con las tablas Grupos.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
Id	Identificador de Usuario	int		Not null	✓	
grupo_id	Identificador perteneciente a Grupos	int		Not null		✓
nombre	Nombre(s) del usuario	varchar	200	Null		
paterno	Apellido paterno del usuario	varchar	200	Null		
materno	Apellido materno del usuario	varchar	200	Null		
correo	Correo del usuario	varchar	200	Null		
activar	Activar o desactivar al usuario	int		Not null		
username	Nombre del usuario	char	50	Null		
password	Contraseña del usuario	char	50	Null		
passwordresp	Respaldo de la contraseña del usuario	char	200	Null		
created	Fecha de creación	date		Null		
modified	Última modificación	date		Null		

TABLA AROS.- Tiene relación con las tablas: Acos y Aros_Acos.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
Id	Identificador de Aro	int		Not null	✓	
parent_id	Nodo padre del árbol	int		Null		
model	Tabla de donde viene el Aro	varchar	255	Null		
foreign_key	Id del Aro en la tabla de donde proviene	int		Null		
alias	Otro nombre del Aro	varchar	255	Null		
Lft	Índice izquierdo	int		Null		
rght	Índice derecho	int		Null		

TABLA ACOS.- Tiene relación con las tablas: Aros y Aros_Acos.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
Id	Identificador de Aco	int		Not null	✓	
parent_id	Nodo padre del árbol	int		Null		
model	Tabla de donde viene el Aco	varchar	255	Null		
foreign_key	Id del Aco en la tabla de donde proviene	int		Null		
alias	Otro nombre del Aco	varchar	255	Null		
Lft	Índice izquierdo	int		Null		
rght	Índice derecho	int		Null		

TABLA AROS ACOS.- Tiene relación con las tablas: Aros y Acos.

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave externa
Id	Identificador de Aro_Aco	int		Not null	✓	
aro_id	Identificador perteneciente a Aros	int		Not null		✓
aco_id	Identificador perteneciente a Acos	int		Not null		✓
_create	Control de creación	char	2	Not null		
_read	Control de consulta	char	2	Not null		
_update	Control de actualización	char	2	Not null		
_delete	Control de eliminación	char	2	Not null		

4.2 DESARROLLO DE SOFTWARE

En un principio, el sistema se instaló localmente en una laptop, es decir, en:

<http://localhost/nicunam/>

Por lo que se instaló y utilizó XAMPP, el cual es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

La versión utilizada fue XAMPP 1.7.3 para Windows Vista Ultimate, misma que consta de los siguientes componentes:

- Apache 2.2.14 (IPv6 habilitado) + OpenSSL 0.9.8i
- MySQL 5.1.41 + PBXT engine
- PHP 5.3.1
- phpMyAdmin 3.2.4
- Perl 5.10.1
- FileZilla FTP Server 0.9.33
- Mercury Mail Transport System 4.72

La base de datos

Se creó la base de datos en MySQL con el motor de almacenamiento InnoDB y se estableció la integridad referencial mediante las llaves foráneas.

Funcionalidad

La aplicación del MVC así como de la metodología RUP, ha sido clave en el desarrollo del sistema ya que se ha podido lograr en principio la funcionalidad parcial y posteriormente la funcionalidad total de éste, dejando para los últimos ciclos la cuestión de la interfaz, que si bien es determinante, no afecta el hecho de separarlos y de esta forma se puede ver de manera práctica cómo se combina por un lado la cuestión de la metodología de todo el sistema junto con el patrón de diseño para brindar comodidad al desarrollador, espacio para el cliente y oportunidad para la calidad en cada una de las etapas.

Para poder aplicar el MVC, se utilizó CakePHP que como ya se había mencionado anteriormente, es un framework de desarrollo de aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails, libre, de código abierto, que implementa el patrón MVC. La versión instalada fue la 1.2.5

❖ *Modelo*

En este punto se hizo la correspondencia entre la base de datos y el modelo, definiendo cada uno de éstos en la aplicación, es decir, un modelo para cada tabla y estableciendo en ellos las reglas del negocio establecidas por el cliente, que si bien ya habían sido tomadas en cuenta durante la creación de la base de datos, aquí se pudieron puntualizar de forma más precisa algunas de ellas.

❖ *Vista - Controlador*

También se trabajó en los controladores definiendo las funciones que se realizan en cada uno de ellos cada vez que se solicita en las vistas, en éstas sólo se colocaron los elementos necesarios que permitieran visualizar tanto los elementos para enviar una solicitud como el resultado de las peticiones hechas, con el fin de observar que todo estaba perfectamente bien conectado y que la salida era correcta, ya que posteriormente se trabajaría en el diseño.

Establecer un sistema funcional requirió de estar realizando muchas pruebas e ir con el cliente a presentarlo cada vez que se tenía una nueva versión para que lo aprobara, y no siempre se lograba el objetivo, sino más bien, existían modificaciones o se añadían requerimientos. Lo cual hizo que el desarrollo del sistema se estableciera en esta etapa durante un periodo mayor de tiempo de lo contemplado.

El diseño

Para poder lograr el diseño de la interfaz del sistema, se integró la librería YUI (Yahoo User Interface Library), o bien, Biblioteca de interfaz de usuario de Yahoo dentro de cakePHP. Y de esta forma, se trabajó en la parte de las vistas de una manera más peculiar.

Con el uso de la librería YUI versión 2 se realizó lo siguiente:

- El menú principal.
- El diseño de las pestañas.
- La presentación de las tablas.
- La organización de las columnas (ordenarlas).
- Los efectos de color en las filas.
- La paginación.
- La búsqueda de datos.

También cabe mencionar que el uso de JavaScript y Ajax fue importante en la aplicación ya que por mencionar algo al respecto, en algunas partes se utilizaron para no tener que abrir una nueva pestaña o tener que cargar la página nuevamente, ahorrando así tiempo y de esta forma brindar un mejor desempeño además de que los efectos logrados son detalles de calidad; un ejemplo de ello se puede ver claramente reflejado cuando se requiere agregar algún registro.

Con la integración de las herramientas, se pudo llevar a cabo lo que se había planteado en la propuesta previamente dada logrando un diseño que además de ser funcional, es muy amigable, vistoso e intuitivo.

4.3 IMPLEMENTACIÓN

El sistema y la base de datos fueron instalados en un servidor proporcionado por el área de servidores del TIC con las siguientes características de software:

- Sistema operativo OpenBSD
- Manejador de bases de datos: MySQL
- Servidor web: Apache
- PHP

Debido a que el sistema se encuentra disponible en línea (vía web), sólo se necesita una conexión a internet ya sea mediante modem, Wi-Fi, LAN, etc. y los requerimientos mínimos suficientes para soportar dicha conexión, en cualquier sistema operativo y contar con un navegador web.

La dirección donde se encuentra alojado es:

<http://www.nic.unam.mx/nicunam>

Una vez instalado el servidor se han realizado las siguientes pruebas:

4.4 PRUEBAS

A continuación se presentan las pruebas que se realizaron al sistema durante la etapa de desarrollo e implementación del mismo:

Caso de Uso	Caso de prueba	Resultado
Entrar	Escribir la dirección URL del sitio WEB.	Ver la página principal del sistema.
Autenticar	Ingresar una clave y/o contraseña no válida.	El sistema mandará un mensaje de error, indicando que el nombre de usuario y/o contraseña incorrecta.
	Ingresar una clave y contraseña correctas.	El sistema mostrará la página de inicio con las funcionalidades extras de acuerdo al usuario que esté ingresando.
Consultar Registros	Buscar información que no existe en la base de datos.	El sistema notificará que no se encontraron registros.
	Recorrer los registros en cualquier dirección.	El sistema desplegará la información de lo que se le solicite.
Consultar Información del Registro	Seleccionar cualquier columna a excepción de las columnas editar y borrar.	El sistema mostrará la información referente a la selección.
Agregar Registro	Ingresar datos válidos correspondientes al tipo de datos.	El sistema no mostrará ningún error en la pantalla relacionado al tipo de dato.
	Ingresar datos que no correspondan al tipo de datos especificado.	El sistema mostrará un mensaje acerca del tipo de datos que debe introducirse en cada campo con error.
	No ingresar datos obligatorios.	El sistema notificará que hay campos obligatorios sin llenar y no se podrá agregar el registro hasta que se llenen.
	Llenar todos los campos y seleccionar cancelar.	El sistema cancelará dicha acción de agregar, borrando la información introducida en cada uno de los campos y quitando la visualización en pantalla.
Modificar Registro	Borrar campos obligatorios.	El sistema notificará que hay campos obligatorios sin llenar y no se podrá actualizar el registro hasta que se llenen dichos campos.
	Ingresar datos que no correspondan al tipo de datos especificado.	El sistema mostrará un mensaje acerca del tipo de datos que debe introducirse en cada campo con error.

Tabla 4.1 Plan de pruebas al sistema de bases de datos (1 de 2).

Caso de uso	Caso de prueba	Resultado
Eliminar Registro	Eliminar registros.	El sistema siempre pedirá confirmación de la eliminación de cualquier registro y lo llevará a cabo de inmediato.
	Cancelar eliminación de registros.	El sistema cancelará la acción de eliminar y automáticamente se seleccionará la fila correspondiente al último movimiento.
Salir	Cerrar sesión	El sistema mostrará un mensaje que ha salido de su cuenta y después redireccionará a la página principal del sistema.

Tabla 4.1 Plan de pruebas al sistema de bases de datos (2 de 2).

4.5 MANTENIMIENTO

Se han insertado en la base de datos todos los registros que ya se tenían, se han actualizado gran parte de ellos y se han agregado más.

Una vez almacenada toda la información en la base de datos, se realizaron nuevamente las pruebas de la tabla 4.1. Logrando ver un buen funcionamiento y desempeño, corroborado con la comparación de los datos contenidos en la base de datos obtenidos de manera manual haciendo las consultas correspondientes de manera directa.

Considero que la mejor manera de probar el sistema es utilizándolo, lo cual se ha hecho desde el momento de su instalación ya que a diario se realizan altas, bajas, cambios y consultas, es decir, se encuentra trabajando al “24x7” como se había planeado desde un principio y no ha presentado ningún tipo de problemas a la fecha, por lo que se puede decir que ya se ha pasado a la etapa de mantenimiento.

CONCLUSIONES

El Sistema de Bases de Datos para las Redes de Voz y Datos de la UNAM cumple en su totalidad con los requerimientos establecidos en un principio y además se han incluido servicios adicionales que complementan dicho sistema.

El sistema fue creado para el Centro de Información de la Red (NIC-UNAM), el cual se encarga de su administración y mantenimiento. Brinda sus servicios principalmente a los usuarios del Centro de Operación de la Red integrado por: NIC-UNAM, NOC-UNAM y TAC-UNAM. Asimismo, a personal de la Dirección de Telecomunicaciones Digitales y la subdirección correspondiente a la que pertenece, es decir, la Subdirección de Redes y Comunicaciones. También está disponible para usuarios del área de seguridad informática. De igual forma, está destinada a los administradores de red pertenecientes a RedUNAM que lo soliciten y/o que cumplan con ciertos requisitos establecidos por NIC-UNAM.

Actualmente se encuentra instalado en un servidor en la Dirección General de Cómputo y de Tecnologías de Información de la Universidad Nacional Autónoma de México (TIC-UNAM) (antes Dirección General de Servicios de Cómputo Académico DGSCA-UNAM).

Disponible en la siguiente dirección:

<http://www.nic.unam.mx/nicunam>

Se encuentra trabajando al 24x7 y no ha presentado problemas de ningún tipo.

El Sistema de Bases de Datos ha permitido agilizar las funciones que se realizan dentro de NIC-UNAM de manera eficaz logrando así un mejor desempeño en la administración a cargo de este Centro. Su interfaz ha resultado ser muy amigable para los usuarios, quienes al consultar, introducir, modificar y eliminar información de la base de datos lo pueden hacer de manera fácil, rápida y segura con el sólo hecho de contar con una computadora y conexión a Internet, no importando la hora o el lugar donde se encuentre el usuario.

La estructura interna de la base de datos es totalmente escalable y portable, lo cual resulta de suma importancia ya que RedUNAM crece día con día. Y debido a que la aplicación se basa en el patrón MVC, se puede separar el modelo, la vista y el controlador, de tal forma que las modificaciones correspondientes a cada una de estas partes se realizan de manera independiente sin afectar a las otras, lo cual es un beneficio del cual se puede hacer uso en todo momento para atender cualquier necesidad que se presente o realizar modificaciones.

Todo lo anterior indica que la creación del Sistema de Bases de Datos para las redes de Voz y Datos de la UNAM se ha concluido de manera exitosa, logrando el objetivo principal y otros objetivos planteados, brindando un plus, lo cual eleva la calidad del producto y lo ubica como una herramienta integral para el uso satisfactorio de los usuarios de este sistema.

Se necesitó de la aplicación de conocimientos teóricos así como de cuestiones prácticas, algunas de las cuales se adquirieron de manera especial para la construcción este proyecto. Respecto a la duración en la realización del mismo, cabe mencionar que a pesar de que las metodologías existentes y específicamente la metodología RUP, la cual fue utilizada en este proyecto, establecen ciertos parámetros, depende de manera directamente proporcional a los riesgos con los que se inician así como el personal involucrado para los distintos roles. Pero en este caso todas las actividades tales como: diseño, programación, documentación, etc., estuvieron ejecutadas por una sola persona, lo cual modifica dichos tiempos.

Trabajar con software libre fue una experiencia gratificante para mí ya que al utilizarlo comprobé que lo importante es conocerlo y saber usarlo para lo cual sólo se necesita disposición y empeño para aprenderlo ya que su difusión y uso ha aumentado de manera significativa en los últimos años y se puede lograr lo mismo que con el software propietario.

Considero que el planteamiento del problema fue crucial y de las etapas en las que se invirtió más tiempo ya que a pesar de haber interactuado con la base de datos anterior, no se entendían bien cada una de las partes por lo que se tuvo que visitar las distintas áreas involucradas del Centro de Operación de la Red para platicar con el cliente las inquietudes surgidas pero existía el inconveniente de que cada vez que se tenían estas juntas se sumaban más requisitos a los ya solicitados.

Existen muchos caminos diferentes para llegar a un mismo lugar pero siempre se tiene que elegir sólo uno de ellos. Y lo mismo pasa con un sistema, existen muchas formas y herramientas para poder crearlo pero depende de la elección que se haga de éstas lo que nos permitirá que el desarrollo sea fácil o complicado, rápido o lento, entre otras cosas, por lo que es de suma importancia tomar en cuenta todos los factores involucrados y realizar un análisis exhaustivo que nos ayude a tomar la mejor decisión en cada caso ya que todos los sistemas son diferentes.

En la actualidad, este tipo de sistemas es lo que se utiliza para el control y la administración de la información, ya que haciendo uso de la tecnología, Internet y las bases de datos se pueden construir sistemas y aplicaciones que respondan a las necesidades específicas de un cliente. Es por esta razón que las tecnologías de la información y comunicación juegan un papel primordial tanto en el desarrollo de una organización como del país entero. Son parte de nuestra realidad al utilizarlas en la vida cotidiana en la mayoría de los lugares que visitamos, a veces de manera consciente y otras sin darnos cuenta pero de alguna u otra forma están presentes y el compromiso ante esta demanda de uso de sistemas como el que se desarrolló en esta tesis es tener bases teóricas firmes, poseer los conocimientos necesarios y estar siempre al día y a la vanguardia en las herramientas a utilizar para el desarrollo de dichos sistemas. Dentro de tales herramientas, aquellas que son software libre han tomado gran fuerza y han evolucionado de manera importante al grado que ahora son

utilizadas en grandes proyectos y en empresas reconocidas a nivel mundial con la seguridad de que en ellas se pueden encontrar los mismos beneficios que en las de software propietario y se cuenta con el respaldo de toda la comunidad mundial de desarrolladores a través de Internet.

Sin lugar a dudas, el hecho de implementar un proyecto que se ha diseñado y desarrollado con tanto interés y esfuerzo es la mejor recompensa que se obtiene ya que todo aquello que se puso en papel ahora es toda una realidad y puede ser utilizado por muchas personas; además, con este Sistema se puede observar claramente lo que es la Ingeniería y cómo se aplica, en este caso, en la creación de sistemas para el beneficio de la sociedad, por lo que trabajar en mi tesis ha sido una gran experiencia que me ha dejado una gran satisfacción y muchos conocimientos tanto teóricos como prácticos que han contribuido a mi desarrollo y crecimiento profesional y personal.

---ANEXOS---

ANEXO A

MANUAL DE USUARIO DEL SISTEMA DE BASES DE DATOS PARA LAS REDES DE VOZ Y DATOS DE LA UNAM (REDUNAM)

ÍNDICE

1. INICIO DE SESIÓN	160
2. OBJETOS DE CAPTURA Y MENSAJES DEL SISTEMA	162
3. DISTRIBUCIÓN DEL SISTEMA	163
4. DEFINICIÓN DE ROLES	164
5. PROCEDIMIENTO PARA CERRAR SESIÓN	165
6. PROCEDIMIENTO PARA OBTENER AYUDA	165
7. PROCEDIMIENTO PARA CONSULTAR REGISTROS	166
7.1 CON BÚSQUEDA GENERAL	166
7.2 CON BÚSQUEDA AVANZADA	167
8. PROCEDIMIENTO PARA AGREGAR REGISTROS	168
9. PROCEDIMIENTO PARA LA EDICIÓN DE REGISTROS	171
10. PROCEDIMIENTO PARA BORRAR REGISTROS	172
11. PROCEDIMIENTO PARA LA ADMINISTRACIÓN DE USUARIOS	173
11.1 GRUPOS	173
11.2 USUARIOS	175

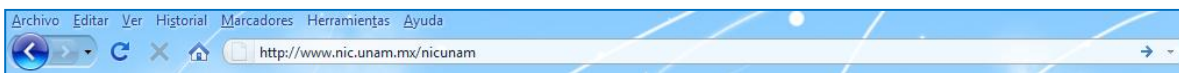
OBJETIVO

En el presente manual se describen las funciones que un usuario registrado puede realizar en el Sistema de Bases de Datos para las Redes de Voz y Datos de la UNAM (RedUNAM) administrada por NIC-UNAM, con el objetivo de brindar información detallada acerca del funcionamiento y de los procedimientos de dicho sistema.

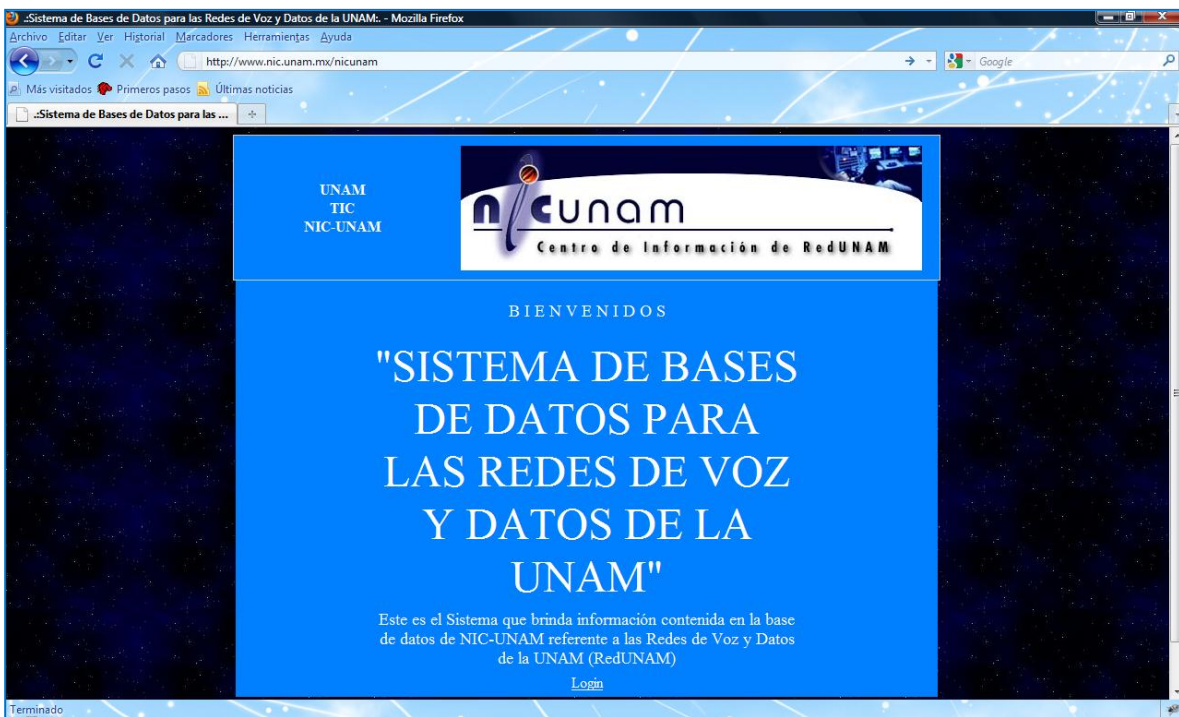
1. INICIO DE SESIÓN

Para poder ingresar al sistema, será necesario realizar los siguientes pasos:

- 1) Especifique en su navegador en la sección de dirección o URL la dirección del servidor en donde se encuentra instalado el sistema:



- 2) El sistema presentará la página principal con la opción de entrar:



- 3) Al seleccionar la opción de entrar, el sistema presentará la pantalla de autenticación donde se debe ingresar la clave de usuario y la contraseña correspondiente. Posteriormente se debe presionar el botón “Entrar” para poder iniciar sesión:



Acceso

Clave: coral

Contraseña: ●●●●

Entrar

Recuerde diferenciar entre mayúsculas y minúsculas

Sistema de Bases de Datos para las Redes de Voz y Datos de la UNAM
UNAM - Dirección General de Cómputo y de Tecnologías de Información y Comunicación (TIC UNAM) - Centro de Información de RedUNAM(NIC-UNAM)

- 4) Veremos la página de inicio del sistema, tal como se muestra en la siguiente imagen:



Sistema de Bases de Datos para las R... | Universidad Nacional Autónoma de México | UNAM | n e unam | Centro de Información de RedUNAM | DIRECCIÓN GENERAL DE CÓMPUTO Y DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN

Ayuda | Cerrar sesión

Menú principal

- Nodos
- Dependencias
- Contactos
- Equipos
- Administración
 - Grupos
 - Usuarios

Nodos

Agregar Nodo

<< primera < anterior 1 siguiente > última >> 15

1 - 4 registros de 4 en total

ID	Nombre del nodo	Fecha de creación	Última modificación	Editar	Borrar
3	TIC	13-07-2010	12-11-2010	Editar	Borrar
4	ARQUITECTURA	13-07-2010	13-07-2010	Editar	Borrar
5	ZONA CULTURAL	13-07-2010	11-02-2011	Editar	Borrar
6	IIMAS	20-07-2010	20-07-2010	Editar	Borrar

Copyright © 2011 Coral Alejandra Montes Carmona. Todos los derechos reservados

2. OBJETOS DE CAPTURA Y MENSAJES DEL SISTEMA

Dentro de las pantallas que componen al sistema se presentarán diversos objetos o controles que permitirán la captura, consulta, modificación y/o eliminación de la información, los cuales tienen las siguientes características:

Objetos	Descripción	Imagen de ejemplo
Cuadros de texto	Son áreas donde se captura la información por parte del usuario	
Ligas	Permiten al usuario navegar a otra área del sitio	
Botones de radio	Objeto de captura que presenta dos posibles opciones de las cuales sólo se podrá elegir una.	
Botones	Son objetos que realizan acciones como: Aceptar, Buscar, Cancelar, Agregar,	
Listas desplegables	Objeto de captura que muestra una serie de valores únicos que puede tomar un campo en específico.	

Tabla 1. Controles de captura

También existen mensajes que el sistema manda para diversas acciones, los cuales se identifican con los siguientes símbolos:



Manda un mensaje de aviso de alguna acción realizada o por hacer.



Envía un mensaje de pregunta para confirmar algo.

Los mensajes de error aparecerán en color rojo al lado del campo donde ocurrió el error, indicando el problema presentado. Esto ocurre cuando se introducen caracteres que no coinciden con el tipo de dato requerido en algún campo, o bien, aquellos campos obligatorios que no se han llenado, ejemplo:

Dependencia: FACULTAD DE INGENIERIA

Tipo de Red: Datos Voz Campo obligatorio

Identificador de Red: hola Por favor ingrese una dirección IP válida

Máscara: Campo obligatorio

Gateway: Campo obligatorio

Inicio de Rango: ocho Por favor ingrese un número entre 1 y 254

Fin de Rango: Campo obligatorio

Interfaz física: V35

Clasificación de Red: LAN WAN Campo obligatorio

Observaciones:

3. DISTRIBUCIÓN DEL SISTEMA

La página de inicio consta de varias secciones que contienen elementos distribuidos como se muestra a continuación:

The screenshot displays the RedUNAM system interface. At the top, there is a header with logos for UNAM, RedUNAM, and the Dirección General de Cómputo y de Tecnologías de Información y Comunicación. Below the header, there is a search bar and a main menu on the left. The main content area shows a table of nodes with the following data:

ID	Nombre del nodo	Fecha de creación	Última modificación	Editar	Borrar
3	TIC	13-07-2010	12-11-2010	Editar	Borrar
4	ARQUITECTURA	13-07-2010	13-07-2010	Editar	Borrar
5	ZONA CULTURAL	13-07-2010	11-02-2011	Editar	Borrar
6	IIMAS	20-07-2010	20-07-2010	Editar	Borrar

At the bottom of the interface, there is a copyright notice: Copyright © 2011 Coral Alejandra Montes Carmona. Todos los derechos reservados.

Área de logos

En esta sección se encuentran los siguientes logos:

(De izquierda a derecha) Logo de la UNAM, logo de NIC-UNAM, logo de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación.

Menú principal

En esta sección se encuentran las opciones principales del sistema presentadas en un menú vertical que se contrae y se expande, mismo que estará disponible en todo momento.

Área de trabajo

Esta área es donde se realiza la mayor parte de la interacción con el usuario ya que es aquí donde se muestra el resultado de la elección correspondiente a las opciones del menú principal, o bien, es la salida para una búsqueda general. A su vez, en ella podemos hacer las distintas acciones tales como: búsquedas avanzadas, consultas, altas y bajas. Asimismo, la administración concerniente a los usuarios.

Área de pestañas

En esta sección se encuentran las pestañas que se van abriendo de acuerdo a la(s) selección(es) correspondiente(s) por parte del usuario, las cuales simulan carpetas formando así un árbol jerárquico. Por lo que cabe mencionar que se pueden llegar a

distintos niveles que van de arriba hacia abajo, de tal forma que mientras más específico sea un recorrido, más niveles estarán abiertos y en cada uno de ellos se tendrá determinado número de carpetas (pestañas) abiertas que varían de acuerdo a las selecciones realizadas.

Área de búsqueda

En esta parte se podrán realizar las búsquedas generales, en cualquier nivel que se encuentre el usuario siempre y cuando dicho nivel cuente con esta opción.

Área de otras opciones

Aquí se encuentran 2 opciones:

1) Ayuda

Cuando el usuario selecciona esta opción, se tendrá acceso al presente manual de usuario con el objetivo de brindar información acerca del sistema en caso de que se tenga alguna duda sobre el funcionamiento, o bien, ampliar dicha información.

2) Cerrar sesión

Al elegir esta opción, se sale del sistema por completo de manera correcta y segura.

Área de copyright

En esta sección se encuentran los créditos y datos correspondientes al autor del sistema.

4. DEFINICIÓN DE ROLES

En el sistema están definidos 3 tipos de roles, los cuales son: usuario de consulta, usuario de edición y administrador. Los cuales pueden realizar ciertas actividades de acuerdo a los permisos establecidos para cada uno de ellos. A continuación se describen puntualmente:

Usuario de consulta.- Es aquel que entra al sistema con el fin de obtener información específica contenida en la base de datos. Sólo tiene permiso de lectura, es decir, sólo puede consultar la información de la base sin modificarla.

Usuario de edición.- Este usuario además de consultar la información, puede modificar los datos contenidos en la base mediante la edición de éstos, o bien, insertar nueva información agregando nuevos registros. También podrá consultar la bitácora del sistema.

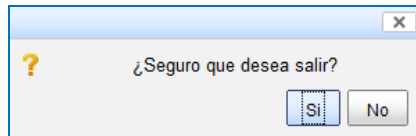
Administrador.- El administrador tiene todos los privilegios sobre la base de datos: consultar, agregar, modificar y eliminar la información contenida en ella; además, está a cargo de la seguridad, del monitoreo de la bitácora y de la administración de los usuarios (mediante la creación de grupos y usuarios con sus respectivas contraseñas las cuales proporcionará a las personas que tendrán acceso). Asimismo, posee todo el control sobre el

sistema en cuanto a diseño y estructura se refiere de tal forma que cualquier cambio interno o externo será realizado por él.

PROCEDIMIENTOS

5. PROCEDIMIENTO PARA CERRAR SESIÓN

Para poder salir del sistema de manera correcta se deberá seleccionar [Cerrar sesión](#), y se mostrará el siguiente mensaje



Al dar clic en “Sí” se cierra la sesión del usuario y se sale del sistema. Al elegir “No”, se cancela dicha acción y el usuario permanece en el sistema

6. PROCEDIMIENTO PARA OBTENER AYUDA

Dar clic en [Ayuda](#) en el área de otras opciones. Se abre en una nueva ventana el presente manual de usuario en formato pdf, donde se puede consultar información concerniente al sistema de bases de datos de RedUNAM y cuenta con la opción de imprimir y/o guardar dicho documento.

IMPORTANTE:

Para poder describir los siguientes procedimientos, se considerarán 2 casos como punto de partida, éstos son:

Caso 1.- Cuando se entra al sistema, se desplegará en el área de trabajo el contenido correspondiente a la pestaña de Nodos.

Caso 2.- Al seleccionar una de las siguientes opciones del menú principal: Nodos, Dependencias, Contactos y Equipos, se abrirá la pestaña correspondiente a dicha selección y en el área de trabajo se mostrará su contenido.

7. PROCEDIMIENTO PARA CONSULTAR REGISTROS

7.1 CON BÚSQUEDA GENERAL

En el área de búsqueda se encuentra un área de texto en donde se debe introducir el “patrón de búsqueda” el cual está formado por caracteres de acuerdo a los siguientes campos en cada caso:

Nivel	Pestaña	Campos
1	Nodos	- Nombre del nodo
	Dependencias	- Nombre de la dependencia - Alias de la dependencia - Siglas de la dependencia
	Contactos	- Nombre del contacto - Clave del contacto - Apellido paterno del contacto
	Equipos	- Ubicación del equipo
2	Switches	- Nombre del switch
	Dominios	- Nombre del dominio
3	Redes Virtuales	- Nombre de la red virtual
	Redes y/o Subredes (de un switch)	- Identificador de red - Tipo de red
4	Redes y/o Subredes (de una red virtual)	- Identificador de red - Tipo de red

Tabla 2. Campos sobre los que se realiza la búsqueda de acuerdo al patrón de búsqueda.

NOTA: No es necesario escribir la palabra completa, basta con proporcionar algunos caracteres si es que no se cuenta con información específica.

Una vez introducido el patrón de búsqueda, se selecciona el botón “Buscar”:

A search input field with the text "DGS" and a button labeled "Buscar".

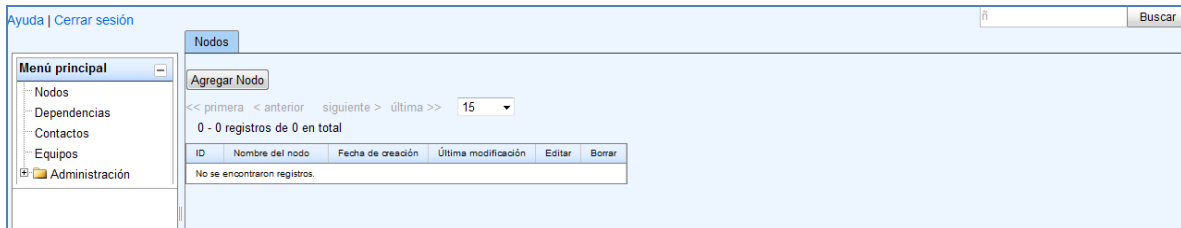
Lo que el sistema realiza en este tipo de búsqueda es comparar el patrón escrito por el usuario con los registros que se tienen en la base de datos para los campos especificados y nos muestra los registros que coinciden con dicha información:

The screenshot shows a web application interface. At the top right, there is a search bar with the text "DGS" and a "Buscar" button. Below the search bar, there is a "Menú principal" on the left with options: Nodos, Dependencias, Contactos, Equipos, and Administración. The main content area is titled "Nodos" and contains a table with the following data:

ID	Nombre del nodo	Fecha de creación	Última modificación	Editar	Borrar
36	DGSCA	19-11-2010	19-11-2010	Editar	Borrar

Navigation controls include "Agregar Nodo", "15" (page number), and "1 - 1 registros de 1 en total".

En caso de no tener éxito se mostrará el mensaje “No se encontraron registros”:



7.2 CON BÚSQUEDA AVANZADA

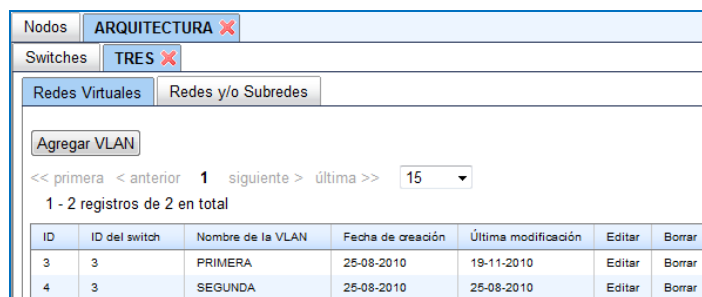
La búsqueda avanzada consiste en ir haciendo una búsqueda cada vez más específica ya que se tiene que ir seleccionando uno de los registros que aparecen cuando se elige una de las opciones del menú principal, o bien, ya estando en las pestañas de los diferentes niveles que se van abriendo (hasta alcanzar el último de los niveles).

IMPORTANTE: Se debe seleccionar cualquier columna excepto la de “Editar” y “Borrar”. Y sólo en el caso de las Redes y/o Subredes la columna “Enlace” tampoco deberá ser seleccionada.

Por ejemplo:

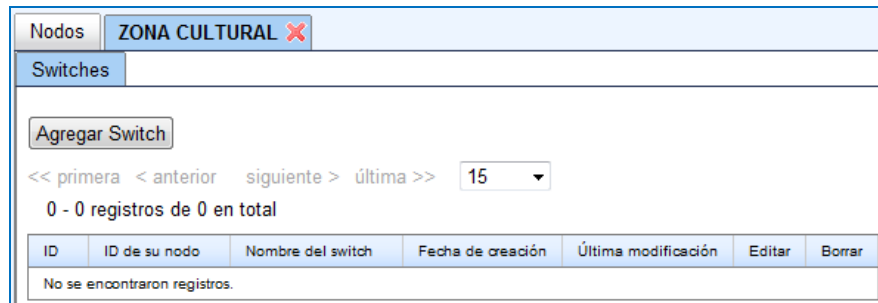
Nodos -> ARQUITECTURA -> Switches -> TRES -> $\left\{ \begin{array}{l} \text{Redes Virtuales} \\ \text{Redes y/o Subredes} \end{array} \right.$

En este ejemplo, en la pestaña Nodos (Caso 1 o 2), se eligió el Nodo ARQUITECTURA y entonces se abrió la pestaña de Switches y en ella se eligió el Switch TRES, que abrió las pestañas Redes Virtuales y Redes y/o Subredes. Ahora se mostrará la imagen correspondiente al ejemplo:



Dentro de la búsqueda avanzada se puede hacer uso de la búsqueda general como herramienta para encontrar la información deseada más rápido.

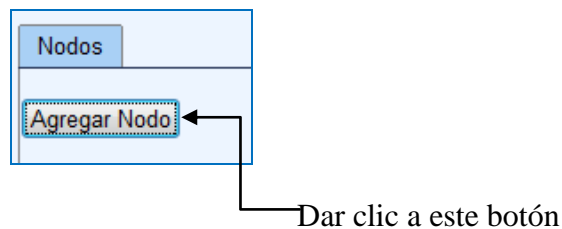
Si al seleccionar un registro, éste no cuenta con registros, se mostrará el siguiente mensaje “No se encontraron registros”



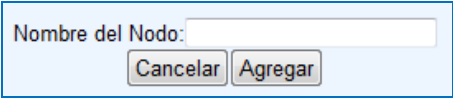
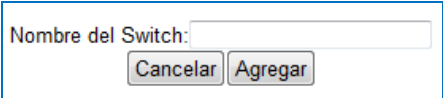
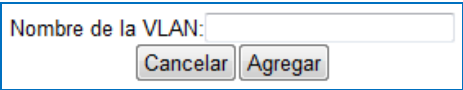
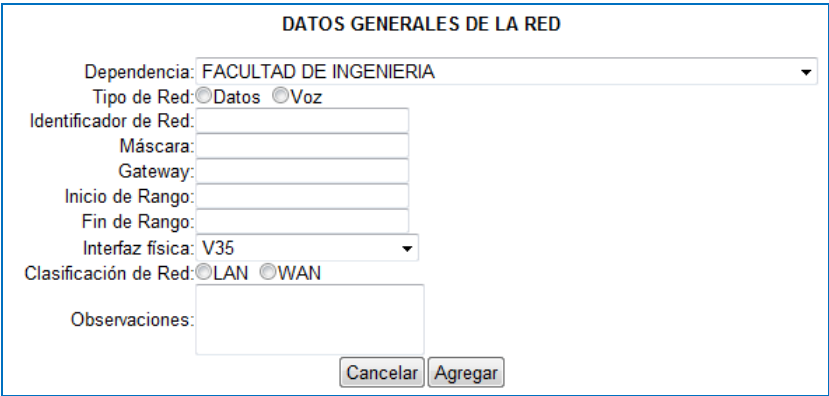
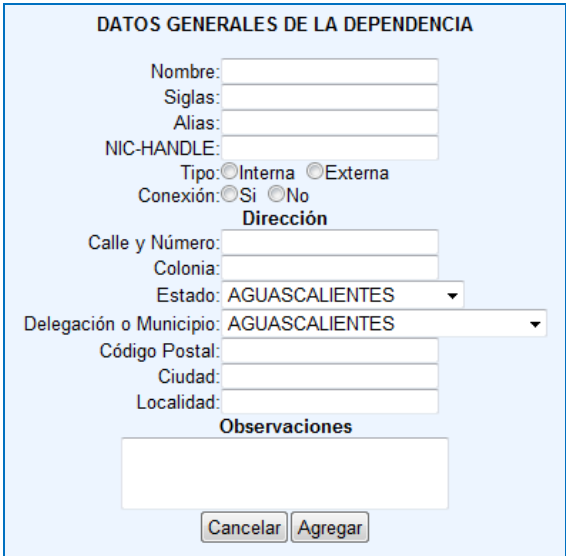
Cabe resaltar el hecho de que el número de pestañas que se abren varía respecto a la pestaña actual donde se encuentren y en algunas pestañas ya no se podrán realizar más búsquedas porque se ha llegado al último nivel, o bien, las pestañas sólo muestran información detallada del registro seleccionado en la pestaña anterior.

8. PROCEDIMIENTO PARA AGREGAR REGISTROS

En cada una de las pestañas, se pueden agregar más registros correspondientes a la pestaña donde se realice dicha acción, para lo cual se debe dar clic en el botón ubicado en la parte superior izquierda, debajo del título de la pestaña, como se muestra en el siguiente ejemplo:



Con lo cual se presentará el formulario a llenar de acuerdo a la pestaña donde se encuentre, el cual consta de varios campos. En la siguiente tabla podemos observar los formularios para agregar registros:

Acción	Formulario
Agregar Nodo	
Agregar Switch	
Agregar Red Virtual	
Agregar Red	
Agregar Dependencia	

Acción	Formulario
Agregar Dominio	<div data-bbox="735 262 1193 363" style="border: 1px solid black; padding: 5px;"> Nombre del Dominio: <input type="text"/> <input type="button" value="Cancelar"/> <input type="button" value="Agregar"/> </div>
Agregar Contacto a la Dependencia	<div data-bbox="735 441 1193 619" style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">DATOS DEL CONTACTO</p> Contacto: <input type="text"/> Tipo de contacto: Técnico <input type="button" value="v"/> <input type="button" value="Cancelar"/> <input type="button" value="Agregar"/> </div>
Agregar Contacto	<div data-bbox="686 693 1243 1213" style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">DATOS GENERALES DEL CONTACTO</p> Título: <input type="text"/> Nombre(s): <input type="text"/> Apellido Paterno: <input type="text"/> Apellido Materno: <input type="text"/> NIC-HANDLE: <input type="text"/> Puesto: <input type="text"/> Correo electrónico: <input type="text"/> Teléfono: <input type="text"/> Celular: <input type="text"/> Observaciones: <input type="text"/> <input type="button" value="Cancelar"/> <input type="button" value="Agregar"/> </div>
Agregar Equipo	<div data-bbox="729 1289 1201 1686" style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">DATOS GENERALES DEL EQUIPO</p> Tributaria: <input type="text"/> Time Slot: <input type="text"/> Puerto: <input type="text"/> DPI: <input type="text"/> Bandwidth: <input type="text"/> Ubicación: <input type="text"/> Observaciones: <input type="text"/> <input type="button" value="Cancelar"/> <input type="button" value="Agregar"/> </div>

Tabla 3. Formularios que presenta el sistema al seleccionar la opción agregar.

Después de llenar los campos correspondientes, se da clic en agregar y podemos observar al nuevo registro en la lista de éstos. En caso de “Cancelar”, se cierra el formulario abierto.

9. PROCEDIMIENTO PARA LA EDICIÓN DE REGISTROS

Para realizar el procedimiento de edición, se deberá tener previamente un registro en la base de datos, el cual se podrá localizar también ya sea por medio de la búsqueda general o la búsqueda avanzada. Una vez ubicado, se debe seleccionar la columna “Editar” tal como se muestra en la siguiente imagen:

Nombre del nodo	Fecha de creación	Última modificación	Editar	Borrar
DGSCA	13-07-2010	05-11-2010	Editar	Borrar
ARQUITECTURA	13-07-2010	13-07-2010	Editar	Borrar
ZONA CULTURAL	13-07-2010	13-07-2010	Editar	Borrar
IIMAS	20-07-2010	20-07-2010	Editar	Borrar

Dar clic aquí

Entonces se abrirá una nueva pestaña en el mismo nivel con los campos que se pueden editar del registro seleccionado, para este ejemplo tenemos que:

Nodos DGSCA ✖

Nombre del Nodo: DGSCA

Después de realizar la edición correspondiente se da clic en Actualizar Nodo:

Nombre del Nodo: TIC

y aparecerá el siguiente mensaje:

Tu información se actualizó con Éxito

Dar clic en “Aceptar”

NOTA: En dado caso de que no se quiera realizar ninguna modificación, sólo bastará con cerrar la pestaña de edición actual abierta.

Por último, se regresa a la página de los registros ya actualizada y vemos el registro modificado:

Nombre del nodo	Fecha de creación	Última modificación	Editar	Borrar
TIC	13-07-2010	12-11-2010	Editar	Borrar
ARQUITECTURA	13-07-2010	13-07-2010	Editar	Borrar
ZONA CULTURAL	13-07-2010	13-07-2010	Editar	Borrar
IIMAS	20-07-2010	20-07-2010	Editar	Borrar

IMPORTANTE: Para los procedimientos de Agregar y Editar, se mostrarán mensajes de error de ser necesario y no se podrá realizar el alta o modificación correspondiente hasta que no se presente ningún tipo de error.

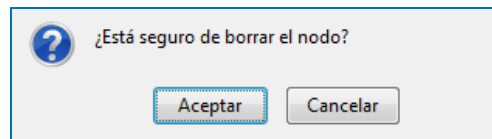
10. PROCEDIMIENTO PARA BORRAR REGISTROS

Para llevar a cabo este procedimiento se deberá tener previamente un registro en la base de datos, el cual podrá ser localizado también por medio de la búsqueda general o la búsqueda avanzada. Una vez ubicado, se debe seleccionar la columna “Borrar” tal como se muestra en la siguiente imagen:

Dar clic aquí

Nombre del nodo	Fecha de creación	Última modificación	Editar	Borrar
TIC	13-07-2010	12-11-2010	Editar	Borrar
ARQUITECTURA	13-07-2010	13-07-2010	Editar	Borrar
ZONA CULTURAL	13-07-2010	13-07-2010	Editar	Borrar
IIMAS	20-07-2010	20-07-2010	Editar	Borrar

Se mostrará el siguiente mensaje de confirmación:



Se da clic en “Aceptar” y el registro es borrado y ya no aparece en la lista:

Nombre del nodo	Fecha de creación	Última modificación	Editar	Borrar
ARQUITECTURA	13-07-2010	13-07-2010	Editar	Borrar
ZONA CULTURAL	13-07-2010	13-07-2010	Editar	Borrar
IIMAS	20-07-2010	20-07-2010	Editar	Borrar

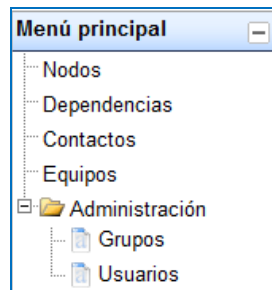
IMPORTANTE: Los registros deberán ser borrados desde el mayor al menor nivel, en caso de ser necesario, ya que de otra forma no procede. Esto se hace para prevenir la eliminación de datos por error y mantener la integridad de los mismos.

En caso de “Cancelar”, no se realiza ninguna acción sobre la base de datos.

11. PROCEDIMIENTO PARA LA ADMINISTRACIÓN DE USUARIOS

Para tener acceso a las funciones de administración de usuarios es importante recordar que sólo el administrador va a poder realizar este tipo de tareas.

En el menú principal se encuentra una carpeta llamada Administración y dentro de ella se encuentran dos opciones: Grupos y Usuarios tal como se muestra en la siguiente imagen:



11.1 GRUPOS

Un grupo se crea a partir de los roles definidos previamente en el sistema, por lo que cada uno de ellos cuenta con ciertos permisos que otorgan privilegios y/o restricciones de acuerdo a las necesidades de los diferentes tipos de usuarios.

Al seleccionar la opción “Grupos” se abre la pestaña correspondiente en el área de trabajo y dentro de ella podemos realizar las siguientes acciones:

- 1) Creación de grupos.- Dar clic en “Registrar nuevo grupo” y se despliega lo siguiente:

Información del grupo (La creación de un grupo tarda varios minutos, ten paciencia)

Nombre:

donde se escribe el nombre del grupo y se selecciona “Guardar”.

- 2) Consulta General.- Ver los registros de los grupos de usuarios existentes en el sistema, los cuales aparecen en una tabla con las siguientes columnas: ID, Nombre, Fecha de creación, Última modificación, Editar y Borrar:

1 - 4 registros de 4 en total					
Id	Nombre	Registro	Última modificación	Editar	Borrar
1	Consultor prueba	00-00-0000 00:00:00	07-09-2010 01:33:36	Editar	Borrar
2	Administrador consulta	00-00-0000 00:00:00	00-00-0000 00:00:00	Editar	Borrar
3	Administrador	00-00-0000 00:00:00	00-00-0000 00:00:00	Editar	Borrar
14	editor	10-11-2010 01:04:27	10-11-2010 01:04:27	Editar	Borrar

- 3) Consulta avanzada.- Se selecciona uno de los registros mostrados en la consulta general, dando clic en cualquier columna excepto las columnas “Editar” y “Borrar”. Y se abre otra pestaña donde se encuentran las funciones y los controladores a los que están asignadas dichas acciones así como su id correspondiente y también los permisos y el id de éstos, de la siguiente forma:

Grupos 14 editor ✖				
<< primera < anterior 1 2 3 4 5 6 7 siguiente > última >> 15 ▾				
1 - 15 registros de 104 en total				
Id	Controller	acción	Permiso_ID	Permisos
3	Pages	display	-1	0
4	Pages	logging	-1	0
6	Contactos	mostrartabcontactos	-1	0
7	Contactos	mostrarcontactos	-1	0
8	Contactos	view	-1	0
9	Contactos	add	-1	0
10	Contactos	edit	-1	0
11	Contactos	delete	-1	0
12	Contactos	logging	-1	0
14	ContactosDependencias	mostrarcontactosdependencias	-1	0
15	ContactosDependencias	view	-1	0
16	ContactosDependencias	add	-1	0
17	ContactosDependencias	delete	-1	0
18	ContactosDependencias	logging	-1	0
20	Dependencias	listado_municipios	-1	0

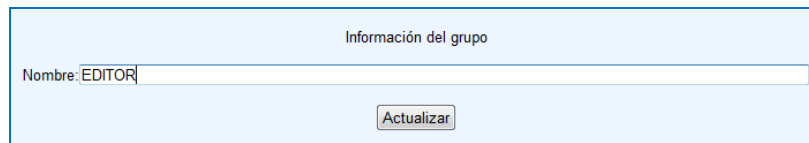
- 4) Configurar permisos.- Dar y/o quitar permisos a las funciones del grupo para tener o no acceso a ellas. Lo cual se hace después de haber realizado una consulta avanzada, dando clic en la columna permisos y seleccionando 1=activar y 0=desactivar, y oprimir “Guardar”:

Procesando cambio de permisos.....				
<< primera < anterior 1 2 3 4 5 6 7 siguiente > última >> 15 ▾				
31 - 45 registros de 104 en total				
Id	Controller	acción	Permiso_ID	Permisos
38	Enlaces	edit	-1	1
39	Enlaces	delete	-1	
40	Enlaces	logging	-1	
42	EnlacesEquipos	index	-1	0

De esta forma se pueden personalizar los grupos asignando permisos.

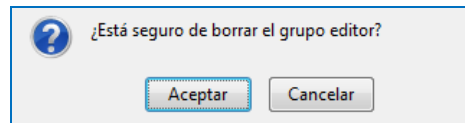
NOTA: Para cancelar, seleccionar “Cancelar”.

- 5) Editar grupos.- Sólo se puede editar el nombre del grupo y esto se hace seleccionando la columna “Editar” del registro que se desee.



Formulario de edición de grupo. El título es "Información del grupo". Hay un campo de texto con el texto "Nombre: EDITOR" y un botón "Actualizar" debajo.

- 6) Borrar grupos.- Se selecciona la columna “Borrar” y aparece el siguiente mensaje:



Y de seleccionar si, se borra el grupo, en caso contrario el grupo se mantiene en la base de datos.

11.2 USUARIOS

Los usuarios son aquellas personas que utilizan el sistema y que están asignados a los grupos previamente creados.

Al seleccionar “Usuarios” en el menú principal se abre la pestaña correspondiente a éstos, donde se pueden realizar las siguientes acciones:

- 1) Agregar usuarios.- Se selecciona “Agregar nuevo usuario” y se muestra el formulario correspondiente para agregar los datos del usuario a registrar.

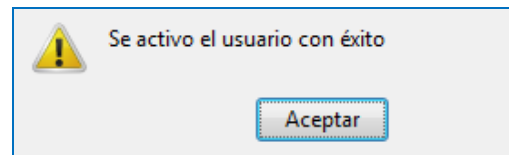
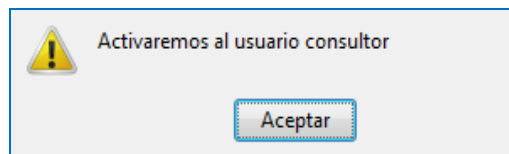
NOTA: Antes de agregar un usuario, se debe saber a qué grupo pertenecerá y en dado caso de que no exista dicho grupo, se deberá crear primero en la sección de grupos y después agregar al usuario o usuarios correspondientes al grupo.

Posteriormente se selecciona Agregar y el usuario es agregado y podemos verlo en la lista de los usuarios registrados.

- 2) Consulta de los registros de usuarios del sistema.- Aparece un listado presentado en una tabla con las siguientes columnas: Activo, ID, Username, Grupo, Cambiar password, Editar y Borrar:

1 - 3 registros de 3 en total						
Activo	Id	Username	Grupo	Cambiar password	Editar	Borrar
<input checked="" type="checkbox"/>	1	consultor	Consultor prueba	Cambiar password	Editar	Borrar
<input checked="" type="checkbox"/>	2	adminconsulta	Administrador consulta	Cambiar password	Editar	Borrar
<input checked="" type="checkbox"/>	3	admin	Administrador	Cambiar password	Editar	Borrar

- 3) Activar usuarios.- Esta opción está presente para que se puedan activar o desactivar a los usuarios registrados en el sistema sin tener la necesidad de eliminarlos permanentemente considerando que en periodos de tiempo, puedan estar activos o inactivos temporalmente, o bien, por cuestiones de mantenimiento. Para ello sólo basta con marcar o desmarcar la casilla correspondiente en la columna “Activar”. Y se muestran los siguientes mensajes:



o bien, los mensajes correspondientes a la inactividad.

- 4) Editar usuarios.- Al seleccionar la columna “Editar” se abre una nueva pestaña para poder hacer modificaciones a los datos del usuario seleccionado:

Usuarios Editar usuario ✕

Información del usuario

Username:

Nombre:

Apellido paterno:

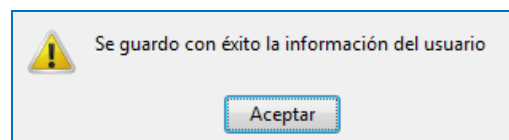
Apellido materno:

Correo:

Grupo:

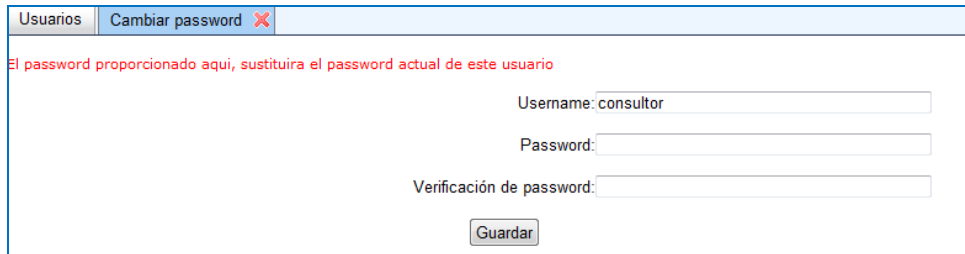
Actualizar

Se muestra el mensaje de confirmación y se regresa a la pestaña de usuarios.



NOTA: Para cancelar, sólo es necesario cerrar la pestaña actual.

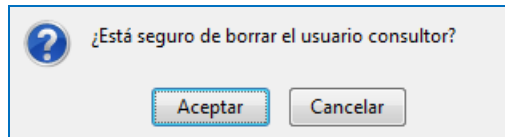
- 5) Cambiar password.- Esta columna sirve para cambiar el password a un usuario y cuando se selecciona se abre una nueva pestaña donde se debe efectuar el cambio:



The screenshot shows a web application window with two tabs: 'Usuarios' and 'Cambiar password' (with a red close button). Below the tabs, there is a red instruction: 'El password proporcionado aqui, sustituirá el password actual de este usuario'. The form contains three input fields: 'Username: consultor', 'Password:', and 'Verificación de password:'. A 'Guardar' button is located at the bottom center of the form.

NOTA: Para cancelar, sólo es necesario cerrar la pestaña actual.

- 6) Borrar usuarios.- Si se selecciona esta columna, se elimina al usuario de manera permanente del sistema. Antes de ello se presenta el siguiente mensaje de confirmación:



The screenshot shows a confirmation dialog box with a question mark icon and the text: '¿Está seguro de borrar el usuario consultor?'. Below the text are two buttons: 'Aceptar' and 'Cancelar'.

ANEXO B

GLOSARIO

API

Interfaz de Programación de Aplicaciones (en inglés Application Programming Interface), es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.

Herramientas de programación para rutinas, protocolos y software.

ASP

Active Server Pages, también conocido como ASP clásico, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS).

CGI

Interfaz de entrada común (en inglés Common Gateway Interface) es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.

DOM

Document Object Model es al mismo tiempo una plataforma y un lenguaje neutral. Proporciona un modelo estándar para conjuntar objetos XML, a la vez que un interfaz estándar para manipular y recorrer los objetos y sus correlaciones.

FRAMEWORK

Un framework en términos de software es un diseño re-usable de un sistema para facilitar la resolución de problemas complejos que presentan comúnmente proyectos o aplicaciones de Software.

HTML

Lenguaje de Marcado de HiperTexto. Es un lenguaje de formato de documentos definido de acuerdo con SGML, es decir, una aplicación de SGML para dar formato a documentos de hipertexto.

HTTP

HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto.

INTELIGENCIA ARTIFICIAL

La inteligencia artificial es el campo de estudio que trata de aplicar los procesos del pensamiento humano usados en la solución de problemas a la computadora.

ISDN

Corresponde a las siglas en idioma inglés para Integrated Services Digital Network, que traducido al español significa Red Digital de servicios Integrados, por lo que se abreviaría RDSI. Estas siglas responden a la denominación de un sistema para las conexiones de teléfonos digitales, especialmente creado para proveer servicios como el envío de voz, de video, así como también, líneas telefónicas digitales o normales que surgen del excedente de los datos simultáneamente.

JDBC

Java Database Connectivity es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

JSP

Java Server Pages es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

ODBC

Open DataBase Connectivity es un estándar de acceso a Bases de datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos almacene los datos.

PATRÓN DE DISEÑO

Un patrón de diseño es una solución probada para un problema en un contexto.

PHP

Es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente PHP Tools, o, *Personal Home Page Tools*). Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

ROBÓTICA

La robótica es el arte y ciencia de la creación y empleo de robots. Un robot es un sistema de computación híbrido independiente que realiza actividades físicas y de cálculo. Están siendo diseñados con inteligencia artificial, para que puedan responder de manera más efectiva a situaciones no estructuradas.

SISTEMAS EXPERTOS

Un sistema experto es una aplicación de inteligencia artificial que usa una base de conocimiento de la experiencia humana para ayudar a la resolución de problemas.

SMIL

Es el acrónimo de *Synchronized Multimedia Integration Language* (lenguaje de integración multimedia sincronizada) y es un estándar del *World Wide Web Consortium* (W3C) para presentaciones multimedia. El lenguaje SMIL permite integrar audio, video, imágenes, texto o cualquier otro contenido multimedia.

SOFTWARE

Software es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo. Software es también conocido como el conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Se trata del conjunto de instrucciones que permite la utilización del computador.

SOFTWARE LIBRE

El software libre también conocido como Open Source software o Free Software, se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

SOFTWARE PROPIETARIO

El software no libre también es llamado software propietario, software privativo, software privado o software con propietario. Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o que su código fuente no está disponible o el acceso a éste se encuentra restringido. En el software no libre una persona física o jurídica (compañía, corporación, fundación) posee los derechos de autor sobre un software negando o no otorgando, al mismo tiempo, los derechos de usar el programa con cualquier propósito; de estudiar cómo funciona el programa y adaptarlo a las propias necesidades (donde el acceso al código fuente es una condición previa); de distribuir copias; o de mejorar el programa y hacer públicas las mejoras (para esto el acceso al código fuente es un requisito previo). De esta manera, un software sigue siendo no libre aún si el código fuente es hecho público, cuando se mantiene la reserva de derechos sobre el uso, modificación o distribución.

SVG

Scalable Vector Graphics es una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados (estos últimos con ayuda de SMIL), en formato XML.

TIC

Tecnologías de Información y Comunicación

URL

Uniform Resource Locator, Localizador Uniforme de Recursos.

WWW

Son las siglas en inglés de “World Wide Web”, también conocida como Web, es el conjunto de documentos interconectados que se encuentran en la red de Internet.

W3C

El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

BIBLIOGRAFÍA

BARCELÓ, José M. et. al. **REDES DE COMPUTADORES**. Eureka Media, España, Primera Edición, 2004, pp. 351

PRESSMAN, Roger S. **INGENIERÍA DE SOFTWARE: UN ENFOQUE PRÁCTICO**. McGraw-Hill, México, Sexta Edición, 2005, pp. 958

SILBERSCHATZ, Abraham; Korth, Henry F.; Sudarshan, S. **FUNDAMENTOS DE BASES DE DATOS**. McGraw-Hill, España, Quinta Edición, 2006, pp. 953

SILBERSCHATZ, Abraham; Galvin, Peter; Gagne. **SISTEMAS OPERATIVOS**. Limusa, México, Sexta Edición, 2002, pp. 840

TANENBAUM, Andrew S. **REDES DE COMPUTADORAS**. Prentice Hall Hispanoamericana, Tercera Edición, México, 1998, pp. 1400

TANENBAUM, Andrew S. **SISTEMAS OPERATIVOS MODERNOS**. Prentice Hall Hispanoamericana, México, 1993, pp. 825

TANENBAUM, Andrew S. **SISTEMAS OPERATIVOS: DISEÑO E IMPLEMENTACIÓN**. Prentice Hall Hispanoamericana, México, 1998, pp. 741

TSAI, Alice Y.H. **SISTEMAS DE BASES DE DATOS: ADMINISTRACIÓN Y USO**. Prentice Hall Hispanoamericana, México, 1990, pp. 607

MESOGRAFÍA

APUNTES DE FICHEROS Y BASES DE DATOS [En línea] disponible en <http://www3.uji.es/~mmarques/f47/apun/apun.html>

LA INGENIERÍA DE SOFTWARE Y RUP [En línea] disponible en <http://www.slideshare.net/dersteppenwolf/la-ingeniera-de-software-y-rup>

SOFTWARE LIBRE VS SOFTWARE PROPIETARIO: VENTAJAS Y DESVENTAJAS
[En línea] disponible en http://gnutransfer.info/descargas/software_libre_vs_propietario.pdf

<http://www.apachefriends.org/es/xampp.html>

<http://book.cakephp.org/es>

<http://cakephp.org/>

<http://www.gnu.org/home.es.html>

<http://www.informatizate.net>

<http://netbeans.org/>

<http://www.php.net/>

<http://www.programacion.com/>

<http://www.proquest.com/>

<http://recursostic.javeriana.edu.co/>

<http://my.safaribooksonline.com/>

<http://www.sybase.com.mx>

<http://www.uml.org/>

<http://www.wikipedia.org>

<http://developer.yahoo.com/yui/>