



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGÓN**

**SISTEMA INTEGRAL PARA ACTIVACIÓN
DE SOFTWARE VÍA INTERNET**

T E S I S

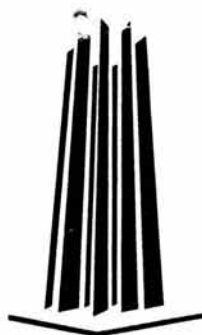
**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

P R E S E N T A N:

MONICA CORTES HERNÁNDEZ

EDUARDO MARTÍNEZ GASCA

ASESOR: ING. BLANCA ESTELA CRUZ LUÉVANO



MÉXICO,

2004



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A G R A D E C I M I E N T O S

MONICA CORTES

A tí papá, Ricardo C. Hdz

Por que con tu trabajo, esfuerzo y cansancio de cada día, lograste sacarme adelante, gracias papito lindo por la educación que me brindaste, por tus consejos y hasta por tus regaños, gracias por tus palabras de aliento y tus buenos deseos, de tí he aprendido muchas cosas, de las cuales he tomado las más importantes para mi vida y que son ahora las que me mantienen en pie para seguir luchando y superandome.

A tí mamá, Ana Maria Hdz. R.

Por haberme dado la vida, por darme tu cariño, comprensión y apoyo, gracias mamita porque no sólo has sido mi mamá sino la mejor amiga que he tenido, juntas hemos reído, hemos llorado, hemos jugado y hasta hemos peleado, gracias por ser parte de mi vida.

A tí máma y a tí papá dedico todos mis logros porque sin su apoyo no hubiera podido salir adelante. Gracias por su amor y cariño, y Gracias a la vida por haberme dado unos padres como ustedes. ¡ LOS QUIERO MUCHISIMO !

A mis hermanos Marisol, Ricardo y Edgar

Gracias por estar presentes en los momentos más importantes de mi vida, así como también por su apoyo en los momentos difíciles, gracias por su cariño y confianza. Los quiero mucho.

A mi mejor Amigo y Esposo, Eduardo Mtz. G.

A tí quiero darte las gracias por todas las cosas buenas y maravillosas que has traído a mi vida, gracias por toda la ayuda, comprensión y apoyo que me has dado, gracias mi Amor por confiar en mí y darme ánimos para seguir luchando y alcanzar mis objetivos. Hemos pasado ya algunos años juntos, y juntos hemos crecido, salido adelante, hemos librado los obstáculos que se nos han presentado en la vida, hemos alcanzado metas y sueños, pero sobre todo hemos echo un gran equipo, no encuentro las palabras exactas para expresarte mi gratitud y mi cariño, pero sabes de antemano que mis logros son tuyos también, que eres y seguiras siendo parte fundamental de mi vida, y que siempre te estaré agradecida por todo lo que has hecho por mí. ¡ TE AMO !

EDUARDO MARTÍNEZ

A mi padres Rubén+ y Eloisa

Gracias por su cariño, en especial a ti mamá por tu apoyo y sacrificios hechos por hacer de mi un hombre de bien, porque has sido parte fundamental en cada uno de mis logros los cuales también son tuyos, nuevamente gracias mamá te quiero mucho.

A mis hermanos Eloisa+, Victor, Ana, Alejandro y Rubén

Mi más sincero agradecimiento por su apoyo, confianza y comprensión, por sus consejos y por su respeto a mi individualidad y proyecto de vida.

A el amor de mi vida Mónica C. H.

Gracias por estar siempre a mi lado en mis triunfos y fracasos, sin tí no hubiera sido posible la realización de este proyecto de tesis, muchísimas gracias por tu cariño, apoyo y comprensión esposa mía. ¡TE AMO!

A mis amigos

Por su amistad, su confianza y su valioso apoyo en mi formación profesional y personal.

AMBOS:

Queremos dar gracias a Dios por habernos dado vida y salud para seguir adelante con nuestros estudios.

A la Asesora Ing. Blanca Estela C. L.

Gracias por su disposición, paciencia, apoyo y consejos que nos brindo para poder llevar a término este trabajo de tesis.

Al Doctor Mark H. V.

Por su amistad, confianza, paciencia y su valioso apoyo en la realización de este proyecto.

Índice de contenido

Introducción	I
Capítulo I. Antecedentes	1
Software	2
Derechos de autor	4
Piratería de software	6
Métodos de protección de software	9
Internet como medio de comunicación	12
Situación actual	14
Ciclo de Vida de software	20
Metodología de desarrollo de software	23
Bases de Datos	35
Capítulo II. Planeación del sistema	46
Objetivo	47
Justificación	47
Propuesta de solución	48
Alcances y limitaciones	50
Estimación de costos	51
Costo de desarrollo	51
Costo de Operación	52
Beneficios del sistema propuesto	53
Capítulo III. Análisis del sistema	54
Determinación de requerimientos	57
Subsistemas que integran el sistema	58
Diagramas de flujo	61
Usuarios del sistema	64
Diagramas de flujo estructurado	65
Estructura de Base de datos Local	68
Diagrama Entidad Relación	72
Estructura de Base de datos de Internet	73
Diagrama Entidad Relación	75

Capítulo IV. Desarrollo del sistema	76
Herramientas para el desarrollo del sistema	77
Diagramas a bloques	82
Subsistema Generador de licencias	85
Subsistema Asistente de registro y activación de software	110
Servidor de licencias de uso de software en Internet	131
Subsistema de registro, obtención y consulta de licencia de uso en Internet	138
Capítulo V. Implantación y Mantenimiento del sistema	149
Pruebas	150
Implantación del sistema	151
Instalación	152
Capacitación a usuarios	154
Revisión del sistema	156
Mantenimiento del sistema	157
Conclusiones	162
Bibliografía	166
Anexos	
Subsistemas en CD	
Generador y administrador de licencias de uso de software	
Asistente de activación de software	
Servidor de licencias de uso en Internet	
Sistema de obtención de licencia de uso en Internet	
Manual de usuario	
Manual de operación	

La propiedad intelectual juega un papel importante en la sociedad, ya que constituye la forma de proteger por una parte a los autores y a todo lo relacionado con la producción de sus obras y por otra parte a los consumidores de dichas obras.

El software como componente principal del ordenador, surge a mediados de la década de los años 60, como integrante del propio hardware, es decir, los fabricantes diseñaban el hardware y elaboraban su propio software, dependiendo de la misma empresa productora. A inicios de los años 80, empezó un mercado independiente de software, surgieron empresas creadoras de programas destinados para computadoras.

En la actualidad el bajo costo del hardware y el alto costo del software original ha generado lo que se denomina "**piratería de software**".

La **piratería de software** es la copia, reproducción, utilización o fabricación no autorizada de productos de software protegidos por las leyes de derechos de autor internacionales.

En promedio, por cada copia autorizada de software informático en uso, se hace al menos una copia no autorizada, o "pirateada" del mismo. En algunos países llegan a hacerse hasta 99 copias no autorizadas por cada copia autorizada en uso.

La piratería de software es un problema mundial con impacto negativo para desarrolladores de software, proveedores, profesionales de software, y consumidores. Gran parte del dinero que se paga al adquirir software original se vuelve a invertir en la investigación y desarrollo de programas más avanzados, a diferencia de lo que sucede cuando se adquiere un software pirata, puesto que el dinero va a un mercado ilegal que no paga impuesto. Cabe mencionar que la industria del Software pierde miles de millones de dólares anualmente a causa de la piratería.

Es por ello que el objetivo fundamental del presente proyecto de tesis es dar una opción más a las empresas de desarrollo de software para proteger contra la piratería el software producido, a través de un sistema integral que permita la activación de software en sólo una ocasión, de tal forma que aunque se realicen copias no autorizadas o se instale el software en más de una computadora, no podrá ser activado. Así se proporciona una mayor protección de los derechos de propiedad intelectual y reducción de la piratería de software.

El presente proyecto comienza haciendo un análisis sobre la problemática actual de las empresas desarrolladoras de software y como la piratería de software se ha convertido en un problema mundial, al final de esta primera parte se explica

el ciclo de vida y la metodología para el análisis que se seguirá durante el desarrollo del sistema integral que presenta este proyecto de tesis el cual esta fundamentado en una base de datos relacional, se hace mención del marco teórico de bases de datos. Una vez determinado lo anterior se prosigue con la planeación del sistema, primer etapa del ciclo de vida, en donde se definirá el objetivo del proyecto, la propuesta de solución, metas y limitaciones que llevarán al éxito y culminación del mismo; así mismo, se realiza un análisis donde se determinan los requerimientos del sistema y se definen los subsistemas que lo integran.

Estos subsistemas son definidos con exactitud en el capítulo 4, por el momento se da una breve descripción de cada uno para proporcionar una idea general del sistema que integran. El primer subsistema se encarga de generar el número de serie que identifica de manera única cada paquete de software a distribuir y así mismo asigna la(s) licencia(s) de uso disponibles que permiten la activación del software contenido en el paquete.

El segundo subsistema consiste en un asistente de registro que es incluido en la aplicación a proteger contra la piratería, esto permite al usuario final registrar y activar el software después de instalarlo; a través del asistente se guía al usuario final durante todo el proceso de registro y activación, para ello el asistente utiliza alguna de las siguientes vías de comunicación:

- Vía Internet contando con una conexión en el mismo equipo
- Vía página Web con conexión a Internet desde otro equipo.
- Vía telefónica con el distribuidor.

El tercer subsistema se encarga de proporcionar al segundo subsistema la licencia de uso para activación de software en el equipo del usuario final, se encuentra en un sitio de Internet y mantiene la información derivada de la activación, así que esta disponible los 365 días del año y las 24 horas del día para que el usuario final no tenga restricciones de tiempo cuando desee registrar y activar su software.

El cuarto y último subsistema, permite el registro y activación de software, comunicándose de forma directa con el tercer subsistema, ello como alternativa cuando el usuario final no disponga de conexión a Internet en el equipo donde realizó la instalación del software.

Una vez definidos cada uno de los subsistemas se comienza con la etapa de diseño, en esta etapa se desarrolla y estructura cada uno de ellos, definiendo diagramas a bloques, algoritmos y forma de interacción con los usuarios finales

Terminada la etapa de diseño pasaremos a la implantación del sistema en la que se integran los subsistemas para su operación y fase de pruebas, comprobando así que el sistema tenga los resultados esperados para que su funcionamiento sea confiable y seguro, finalmente se contempla la capacitación a usuarios.

Antes de finalizar este proyecto se presenta la conclusión a la que se llegó con la investigación y desarrollo del sistema, además de incluir la bibliografía que sirvió de apoyo para fundamentar este trabajo de tesis.

Por último se exponen algunos complementos a este sistema tales como manual de usuario y manual de operación, dichos manuales servirán para que el usuario y el operador aprendan a utilizar de forma correcta el sistema.

Antecedentes

Software

Derechos de autor

Piratería de software

Métodos de protección de software

Internet como medio de comunicación

Situación Actual

Ciclo de Vida del software

Metodología de desarrollo del sistema

Bases de Datos

El software es uno de los valuartes tecnológicos más apreciados de la era de la información puesto que rige el funcionamiento del mundo de las computadoras y de Internet. Desgraciadamente, por este motivo y por la facilidad con la que se pueden crear copias exactas de los programas en cuestión de segundos la piratería de software se encuentra muy extendida.

Se pueden encontrar prácticas de piratería en casas, escuelas, negocios, instituciones gubernamentales, etc. Los piratas informáticos no sólo perjudican a las compañías que fabrican software, sino que también es imposible reinvertir el dinero que éstas obtienen en investigación y desarrollo de programas más avanzados, también perjudican a todos los usuarios. Por este motivo, cualquier forma de piratería de software (incluso una copia de un programa para un amigo) se considera ilegal.

Software

Software es el conjunto de programas que la computadora emplea para manipular datos. Sin el software, la computadora sería un conjunto de medios sin utilizar.

El software es un conjunto de programas, procedimientos, y rutinas asociados con la operación de un sistema de cómputo. Distinguiéndose de los componentes físicos llamados Hardware. Comúnmente a los programas de computación se les llama software; el software asegura que el sistema cumpla por completo con sus objetivos, opere con eficiencia, este adecuadamente documentado y suficientemente sencillo de operar.

El software se clasifica en las siguientes categorías:

- Programas de Sistema
Dentro de esta categoría se mencionan los más importantes que son:
 - Sistemas Operativos
- Lenguajes de Programación
- Software de Aplicación

Programas de Sistema:

Sistemas Operativos: El sistema operativo es, junto con el microprocesador, la parte más importante de un equipo informático. La función de un sistema operativo es básicamente la de establecer la comunicación lógica entre los diferentes elementos de la computadora para que puedan funcionar las diferentes aplicaciones, intentando mejorar la relación entre el usuario y la máquina y optimizando al máximo los recursos de todo el sistema.

Las funciones del sistema operativo dependen de cual se utilice, pero básicamente son las siguientes:

- Planificación, iniciación y supervisión de los programas.
- Asignación de memoria y gestión de los dispositivos del sistema.
- Inicialización y control de operaciones de entrada y salida.
- Control de errores del sistema.
- Coordinar la comunicación entre los elementos del sistema y el usuario.
- Si soporta multiproceso y tiempo compartido, gestionar estas operaciones.

El sistema operativo provee una plataforma de software por encima de la cual otros programas llamados aplicaciones pueden ejecutarse. Los programas de aplicación tienen que crearse de acuerdo a la plataforma en donde se van a ejecutar.

Lenguajes de Programación:

Es un conjunto de símbolos, caracteres y reglas (programas) que le permiten a las personas comunicarse con la computadora. Los lenguajes de programación tienen un conjunto de instrucciones que permiten realizar operaciones de entrada/salida, cálculo, manipulación de textos, lógica/comparación y almacenamiento/recuperación.

Software de Aplicación:

El software de aplicación está diseñado y escrito para realizar tareas específicas personales, empresariales o científicas como el procesamiento de nóminas, la administración de los recursos humanos o el control de inventarios. Todas estas aplicaciones procesan datos (recepción de materiales) y generan información (registros de nómina), para el usuario.

Por otro lado el progreso de la tecnología y la velocidad con la que avanza día a día, implica que muchas veces la tecnología de la que dependen hoy países y empresas pueda quedar obsoleta casi con la misma velocidad. Es por ello que los derechos de propiedad intelectual han sido el marco jurídico adecuado para responder a estos adelantos de la ciencia y la tecnología.

Derecho de autor

Comprende el conjunto de derechos que las leyes reconocen y confieren a los autores y creadores de las obras intelectuales expresadas mediante la escritura, la imprenta, la palabra hablada, la música, el dibujo, la pintura, la escultura, la televisión, la radiodifusión, el grabado, el cinematógrafo, el disco y por cualquier otro medio de comunicación existente. Su objetivo es garantizar la seguridad jurídica de los autores o titulares de derechos conexos y patrimoniales y además, salvaguardar, promover, diseminar y enriquecer el acervo cultural de las naciones.

Es decir, es una forma de protección proporcionada por las leyes para los autores de "obras originales. Esta protección está disponible tanto para obras publicadas como para las obras que todavía no se hayan publicado.

El derecho patrimonial implica la facultad de obtener una justa retribución por la explotación lucrativa de la obra y tiene como contenido sustancial el derecho de publicación, reproducción, traducción, adaptación, ejecución y transmisión. Específicamente, la Ley de Derechos de Autor, dispone de sanciones para las personas u organizaciones que vendan, distribuyan o pongan a disposición de terceros cualquier creación intelectual, sin la respectiva autorización de su dueño, así como al pago de las indemnizaciones.

Ahora bien, las leyes de derechos de autor y las disposiciones de los tratados internacionales sobre los derechos de autor protegen a los programas de cómputo (software) de su copia ilegal. Dichas leyes y tratados protegen los derechos del propietario del software y le otorgan una variedad de derechos exclusivos que incluyen el derecho a su reproducción o duplicado. Consecuentemente, la copia de los programas de software sin la autorización de su autor constituye una violación a sus derechos.

Es ilegal para cualquier persona violar los derechos que le provee el código de los derechos de autor al dueño de los mismos. Estos derechos, sin embargo, no están ilimitados en cuanto a su alcance.

Desde la sección 107 hasta la 120 de la Ley de Derechos de Autor de 1976 se establecen ciertas limitaciones a estos derechos. En algunos casos, estas limitaciones son excepciones específicas de la responsabilidad de los derechos de autor. Una de las mayores limitaciones es la doctrina de "fair use" ¹, la cual está basada en la Sección 107 de la Ley de Derechos de Autor de 1976. En otros casos, la limitación es una forma de "otorgamiento de licencia compulsoria" bajo la cual

¹ Doctrina del uso justo o "Fair Use": Uso imparcial o uso razonable, por ejemplo el derecho a respaldar información para uso personal, derecho a citar partes de obras, entre otros necesarios para permitir un "uso razonable" de la obra.

ciertos usos de trabajos registrados están limitados al pago de derechos específicos y al cumplimiento de las estipulaciones legales.

La protección de los derechos de autor existe desde que la obra está creada de una manera permanente. Los derechos de autor sobre una obra creada se convierten inmediatamente en propiedad del autor que creó dicha obra.

La copia o el uso no autorizado de un programa de computadora en cualquier forma distinta a la permitida por las leyes de derechos de autor o la licencia de software emitida por el fabricante, es lo que se denomina comúnmente como *Piratería de Software*, es decir, un delito legalmente penalizado.

Piratería de Software

Llamamos piratería de software a la realización de copias ilegales de programas originales o la falsificación y distribución de imitaciones de productos de software.

La piratería de software también se refiere a la realización de copias no permitidas, es decir, la piratería de software es la copia, reproducción, utilización o fabricación no autorizada de productos de software protegidos por las leyes de derecho de autor internacionales.

La piratería de software afecta a la economía de un país puesto que pone en peligro aquellos empleos relacionados con la investigación, creación y reproducción de Software.

Las empresas más perjudicadas por la piratería de software son aquellas pequeñas empresas dedicadas a la investigación de nuevos programas y que no poseen un capital muy amplio para cubrir las pérdidas que se pueden producir por causas de la piratería. Existen en la actualidad distintos tipos de piratería:

Tipos de piratería de software

Copia entre usuarios finales:

Consiste en copiar e instalar software sin el número adecuado de licencias. Cuando se compra un software para una máquina, también se compra una licencia para usar dicho software. Este tipo de piratería consiste en instalar dicho software en varias máquinas con el mismo número de licencia en vez de tener un número de licencia para cada máquina.

Instalación en disco duro:

Este tipo de piratería es usado por aquellas empresas que compran un software con su licencia, cuya finalidad es la instalación de dicho software en computadoras que saldrán a la venta con el software instalado.

Piratería a través de medios electrónicos:

Consistente en bajar software de Internet y números de serie que no son los autorizados por la empresa de software a la cual pertenece.

Contrabando:

Este tipo de piratería consiste en realizar copias de seguridad de uno o más software para luego distribuirlas.

Falsificación:

Consiste en la duplicación ilegal y venta de software con derechos de autor, frecuentemente en una forma diseñada para que el producto parezca legítimo. Las falsificaciones de software pueden ser muy sofisticadas e incluir réplicas de paquetes, logotipos y dispositivos contra imitaciones, como hologramas.

Uso incorrecto de la licencia:

Este tipo de piratería ocurre cuando el software se distribuye fuera de los canales de distribución legítimos para los cuales el producto fue diseñado o cuando el producto se utiliza de manera no permitida en el contrato de licencia.

La piratería de software parece algo inofensivo y eficaz, pero puede acarrear consecuencias graves tanto para el usuario como para el que ejerce la piratería.

Las desventajas que puede contener un software ilegal son:

- Virus Informáticos.
- El software puede no traer manuales y demás herramientas para aprender a usarlo.
- La copia pirata puede tener problemas al momento de realizar una actualización.
- Adquirir software pirata puede acarrear problemas legales para el usuario o bien este puede ser sujeto a multas.
- El software no puede ser registrado a nombre del usuario o al de la organización.

Las ventajas que se pueden obtener usando software legal son:

- Los titulares de los derechos de autor (propietarios legítimos del software) de la mayoría de las aplicaciones comerciales otorgan una licencia para que el cliente la utilice de acuerdo a lo establecido en un tipo de contrato comúnmente denominado contrato de licencia de usuario final.
- Documentación original y completa del producto, lo que es indispensable para su correcta configuración, optimización y máximo desempeño.

- Módulos completos y operativos, el producto se adquiere como una sola unidad funcional e integrada, minimizando los riesgos para el usuario.
- Acceso de soporte técnico del fabricante para resolver con rapidez y efectividad problemas eventuales, sin afectar los tiempos de respuesta en los servicios que presta la organización.
- Posibilidad de realizar una copia autorizada con el fin de ser utilizada como respaldo.
- Dependiendo de la negociación con el fabricante o distribuidor autorizado, se puede tener acceso a actualizaciones periódicas del producto adquirido.
- Licenciamiento para el uso del producto, lo que autoriza formalmente a la organización para usar las aplicaciones de acuerdo a las cláusulas establecidas en el contrato y evitar así las delicadas y desagradables situaciones contractuales con el fabricante.
- Consideraciones éticas y de imagen dentro de la organización. Para los funcionarios, clientes y proveedores es importante saber que nuestra organización utiliza productos legítimos para brindarles los servicios que ellos requieren.

Como se ha visto hasta el momento la piratería de software representa un gran problema para las empresas desarrolladoras de software, es por ello que hoy en día existen empresas dedicadas a crear métodos de protección de software.

Métodos de protección de software

Los primeros programas protegidos llegaron del brazo de las computadoras de 8 bits. Al utilizar cintas magnéticas cualquier persona podía copiar fácilmente un programa original. Los programadores pronto idearon sofisticados sistemas turbo, que aumentaban o reducían la velocidad de lectura de los datos, de manera que solo se podían duplicar con equipos de alta calidad. Esto no fue un verdadero problema para los piratas, pues descifraron todos y cada uno de ellos, o utilizaron aparatos que permitían hacer un volcado de memoria en un cassette una vez que el programa original se había cargado evitando así cualquier protección. Con la llegada de los disquetes, las claves se hicieron más sofisticadas. Estas se dividían en tres grupos: las protecciones basadas en software, en hardware y en algún tipo de material impreso en papel.

La protección más común era el "disco llave" normalmente, el primer disco del programa escondía un programa más el sistema de encriptación y formateado a un tamaño no habitual, mediante complejos métodos, para que, al intentar copiarlo hiciera un error de lectura. Este disco debía insertarse cada vez que se quisiera acceder a la aplicación, pero los "hackers"² no tardaron demasiado tiempo en crear ficheros "crack" que se ejecutaban antes que el fichero principal de detección del disco y lo anulaban.

Ante el evidente fracaso de estos sistemas de protección, las empresas han ideado otros métodos de protección más eficaces como son:

Número de serie:

El número de serie/código de autorización es el más fácil de usar, pero también es el más fácil de violar. Hay varios sitios Web que proveen números de serie que se le pueden meter a un programa pirateado para hacer que funcione. Microsoft y FileMaker están entre aquellos que usan este tipo de protección. Algunos programas funcionan en modo de demo hasta que se ingresen los códigos apropiados y entonces se transforman en versiones totalmente funcionales. En este caso, el software puede ser fácilmente transportado de una computadora a otra y puede funcionar en más de una computadora a la vez aunque el acuerdo de licencia puede prohibirlo.

Archivo oculto:

La colocación de archivos ocultos funciona mucho mejor. Se deben ingresar códigos para desbloquear el programa, pero si se formatea el disco

² **Hacker:** Persona que tiene la inteligencia y la capacidad para entrar a cualquier sistema computacional sin estar autorizado. El Hacker aprovecha los agujeros y las fallas de los sistemas de seguridad. Sus intereses se centran en el conocimiento del funcionamiento de los sistemas.

duro o si el programa se mueve a otro disco, el programa pide la reinstalación desde el CD original. Esto lo hace menos atractivo para los piratas debido a que se tiene que circular un CD original para instalar el programa. Se puede instalar el software en más de una computadora y correr múltiples copias si se instalan desde el disco original. (Si el acuerdo de licencia lo permite).

CD original:

Solicitud de inserción del CD original cuando se inicia la aplicación. Algunas aplicaciones se revierten a modo demo si el CD no está presente.

Disco de autorización:

Utilización de discos floppy de autorización. Estos floppys están dañados físicamente con el fin de que no se pueda escribir en algunos sectores. Se pueden hacer copias de los discos, pero se tendría que dañar exactamente el mismo sector y entonces escribir un programa de copia que haga una copia exacta del original, pero que no falle cuando encuentre el sector malo. Si la autorización está en el disco duro, el programa no funciona. Si se instala el software en una computadora distinta, funciona si se inserta el disco de autorización cuando el software lo pide. Sin un disco de autorización, el programa termina su ejecución.

Desafío/respuesta:

La primera vez que se trata de usar el software, el mecanismo de protección mira el disco duro y genera una serie de palabras basadas en el disco duro que se está autorizando. Esta serie de palabras se envían al vendedor de software por E-mail, el cual genera una respuesta y la manda por E-mail. Cuando se ingresa la respuesta correspondiente, el software se activa.

Si se formatea el disco duro, solamente se ingresa nuevamente la misma respuesta y el software se vuelve a activar. Si se quiere mover el software a otra computadora, primero se debe mandar un E-mail al vendedor con la nueva serie de palabras de desafío junto con una explicación convincente de porque se tiene una nueva serie de palabras de desafío.

Dongle:

El dongle es un dispositivo de hardware que puede ser leído desde el software. El dongle contiene el número de serie del software autorizado. Si el dongle no está conectado a la computadora, el software no funciona. Si se cambia de computadora o de disco duro, el software puede ser reinstalado y el mismo dongle activa el software. Cada software con protección de dongle

usa su propio dongle. Por lo cual uno de los inconvenientes de este método es su costo que repercute en el precio final de la aplicación.

Por otro lado, uno de los negocios más florecientes en la actualidad es la piratería, especialmente en Internet. A medida que aumenta el número de usuarios con computadora y el uso de Internet, también aumenta el número de casos de piratería de software. Este último ha sido un factor que contribuye a la piratería debido al gran crecimiento e importancia que ha adquirido.

La piratería de software ha crecido dramáticamente con la expansión del Internet. Los programas cargados en los sistemas de boletines electrónicos (BBS) o de servicios comerciales de comunicación electrónica pueden ser descargados o enviados por correo electrónico a individuos que no tienen una licencia. La piratería a través del Internet es una fuente primaria de virus de computadoras.

Internet como medio de comunicación

Internet se ha convertido en un medio de comunicación, ya que nos permite la comunicación con cualquier parte del mundo ofreciéndonos retroalimentación instantánea si así se requiere, esto gracias a los "chats", o casi instantánea por medio del tradicional "correo electrónico", el cual se puede contestar, consultar y enviar a cualquier hora y en cualquier momento. Así que no se puede negar que Internet ya es un medio de comunicación, que ha venido a complementar e incluso a revolucionar los medios tradicionales de comunicación, causando un gran impacto en lo que se refiere a los medios audiovisuales, ofreciendo de una manera rápida y efectiva información instantánea que viaja por el ciberespacio trayendo al monitor de la computadora imagen y sonido. Es un medio que utiliza el mensaje por escrito, voz e imagen, que aún cuando se puede catalogar de "nuevo", cada día va ofreciendo mayor apertura en la comunicación internacional. A diferencia de la T.V., Internet abre la posibilidad de que el receptor deje de ser pasivo y de esta manera pueda intercambiar información con el emisor.

El dominio de la comunicación de datos se ha convertido en una parte fundamental de los medios de comunicación de las sociedades modernas. Los nuevos criterios de exigencias en materia de comunicación de redes globales y el comportamiento de factores externos a la realidad informática actual, han obligado a las sociedades del mundo entero a disponer de los conocimientos necesarios para navegar en la red y estar al día con el cúmulo de información que allí se maneja. Cada vez más comunidades de todas índoles, financieras, organizacionales, académicas, institucionales reconocen la importancia de este instrumento tecnológico como medio informativo.

Transmisión de Datos en Internet

La gran rapidez con la que Internet se ha expandido y popularizado en los últimos años ha supuesto una revolución muy importante en el mundo de las comunicaciones, llegando a causar cambios en muchos aspectos de la sociedad. Lo que se conoce hoy como Internet es en realidad un conjunto de redes independientes (de área local y área extensa) que se encuentran conectadas entre sí permitiendo el intercambio de datos y constituyendo por lo tanto una red mundial que resulta el medio idóneo para el intercambio de información, distribución de datos de todo tipo e interacción directa con otras personas.

En los últimos años, la red Internet ha ido introduciéndose en nuestra sociedad de un modo enormemente acelerado y profundo. Actualmente, la Red de Redes ya no sirve únicamente para comunicarse o para buscar información. Se está utilizando también para comprar y vender productos, prestar y recibir servicios, establecer relaciones institucionales entre el gobierno y los ciudadanos, acceder a

datos e información sensible, etc. De este modo, hoy se puede presentar la declaración de impuestos, consultar nuestra cuenta bancaria y comprar el último libro de nuestro autor favorito sin movernos de casa gracias a Internet. Todo ello es posible gracias a dos elementos fundamentalmente, que no siempre han ido de la mano pero que se necesitan y complementan: la Tecnología y la Ley: La Tecnología es la base de la "Sociedad de la Información". Gracias al desarrollo de la Informática y de las Telecomunicaciones, así como del software y de los protocolos de comunicación, existe Internet.

La Ley, por su parte, no puede desconocer la injerencia de la Tecnología en la realidad y debe adaptarse a ella de un modo paulatino: en primer lugar, interpretando la normativa tradicional a la luz de las nuevas relaciones y, en segundo lugar, aprobando nuevas leyes que garanticen la Seguridad y el pleno reconocimiento jurídico de las operaciones realizadas en la Red, así como la protección de los derechos de las entidades y de los ciudadanos en este nuevo entorno. En la actualidad, tanto la Tecnología como la Ley están suficientemente desarrolladas como para dotar de una suficiente Seguridad a Internet, así como al Comercio Electrónico. Pero es necesario que ambas disciplinas caminen de la mano para lograr este objetivo común. Ambas se necesitan y, a la vez, se complementan.

Piratería de software en Internet

La piratería de software ha aumentado de forma significativa con la aparición de Internet; anteriormente la piratería se limitaba a medios de almacenamiento físico. Internet ha conseguido poner en línea no sólo el software pirata sino los productos de la industria discográfica y cinematográfica. Los que practican la piratería se sirven de la Red para negociar con productos informáticos (software) o de entretenimiento (música o libros electrónicos), que pueden viajar gratis a través del ciberespacio, saltándose los canales convencionales, y con la garantía de mantener el anonimato.

Se calcula que 1 de cada 2 paquetes de software que se utilizan es ilegal. Más de 17,000 páginas WEB incluyen software pirata y se puede acceder a ellas desde los buscadores, como Geocities, Tripod, Freeyellow y otros servidores gratuitos. Como el famoso buscador de cracks, passwords y downloads: <http://www.astalavista.com> o los sitios web de piratería: <http://start.at/qwert/> y Jolly Roger.

Situación Actual

La situación actual del mercado de software es realmente desafortunada; muchos usuarios atropellan impunemente los derechos de los autores y fabricantes. La actividad ilícita de reproducir los programas de cómputo no solo daña a las compañías desarrolladoras de estos productos, sino también causa cuantiosas pérdidas fiscales para los gobiernos, además de inhibir la generación de miles de empleos nuevos bien remunerados. Así mismo provoca a las empresas que la practican un cuantioso daño económico, moral y de imagen. En la actualidad el mercado del software presenta algunos inconvenientes para el usuario, los siguientes puntos podrían considerarse como factores para que el usuario prefiera comprar una copia pirata de un software y no un producto original.

- No hay un compromiso serio de algunos fabricantes garantizando el soporte técnico oportuno para lograr el funcionamiento correcto del software. Los comentarios en revistas informáticas acerca de los errores en los programas aumentan la inseguridad y la desconfianza en los usuarios.
- El costo de un paquete de informática supera en ocasiones el sueldo mensual promedio. Los descuentos a estudiantes son escasos: un estudiante no puede pagar \$3,000 por un paquete combinado de aplicaciones, ni \$2,000 por un programa de dibujo asistido por computadora. Estos precios fomentan la copia furtiva de aplicaciones entre estudiantes, que aprenden este modo de proceder sin dar importancia alguna a los derechos de los autores.
- Las actualizaciones de los programas no son gratis, ni siquiera las mejoras de las versiones. El usuario que quiera actualizar sus programas tiene que volver a pagar normalmente más de la mitad del producto completo, salvo en el caso de errores muy escandalosos en los que suele distribuirse a un precio muy bajo.
- No se ofrece un servicio de utilerías gratuito relacionadas con el software.
- Precio de la aplicación al usuario: Los niveles de ingreso pueden afectar la capacidad de los consumidores para adquirir software original, su elevado precio es lo que impulsa a muchos a optar por copias ilegales. La motivación para la piratería es el significativo diferencial de precios entre las versiones legales y las piratas. El problema clave es que los precios son fijados en Estados Unidos de Norte América, que son significativamente más altos que lo que pueden pagar las personas en la mayoría de los países. En una gran cantidad de estos, la adquisición

legítima de software no es una opción que la mayoría de los consumidores considere seriamente.

En la ciudad de México es posible encontrarnos con la venta abierta de software donde por 100 pesos puede adquirirse un CD con aproximadamente 10 aplicaciones de diseño gráfico, modelado 2D y 3D, retoque fotográfico, utilerías, fuentes, etc., y en sus versiones más recientes. No cabe duda que estos ofrecimientos resultan ser una gran tentación para los usuarios de cómputo. Gran parte del dinero que se paga al adquirir software original se vuelve a invertir en la investigación y desarrollo de programas más avanzados, a diferencia de lo que sucede cuando se adquiere un software pirata, puesto que el dinero va a un mercado ilegal que no paga impuesto. Esto no incentiva los canales regulares, favorece la competencia desleal, reduce la posibilidad del empleo formal y se deteriora la imagen del país ante la inversión extranjera. Sin un compromiso para reducir la piratería, se impide el potencial para el crecimiento económico mundial.

El alto índice de piratería inhibe el desarrollo de la industria del software y lo imposibilita para alcanzar su completo potencial mundial al engañar a los desarrolladores legítimos de software y a las compañías con los derechos y regalías de la propiedad intelectual. Las estadísticas han demostrado que en algún momento el 69% del mercado de usuarios individuales en países latinoamericanos, manejan programas piratas.

Ante estos graves resultados, autores y productores de estos programas acudieron a las Naciones Unidas a solicitar protección sobre todo para la duplicación no autorizada. Su petición se sustentaba en que el software es un producto de proceso creativo con características de individualidad y por tanto, un bien intelectual susceptible de protección a través de propiedad industrial, derecho de autor u otro derecho especial. Esta solicitud fue trasladada a la Organización Mundial de Propiedad Intelectual (OMPI), quien convocó a un grupo consultivo que decidió la protección jurídica a los programas de ordenador bajo el derecho de autor.

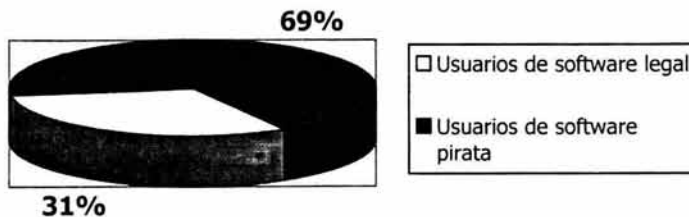


Figura I.1 Usuarios de software en países Latinoamericanos

Sin embargo, a pesar de la existencia de esta protección, la reproducción no autorizada de software está proliferando cada día más, incluso se ha llegado a convertir en una amenaza empresarial en el ámbito internacional. La reproducción informática ilegal, es una práctica muy extendida en las empresas, por lo general suelen comprar un paquete de software, lo cargan en su red y a continuación realizan numerosas copias ilegales que son instaladas en el resto de sus computadoras sin pagar por ello. Con todo esto se puede demostrar que estas empresas, se están aprovechando del sacrificio de miles de personas que dedican esfuerzo humano y económico para crear nuevos productos informáticos en beneficio de la comunidad.

Un factor que contribuye a la piratería de software es la disparidad entre las leyes de propiedad intelectual y derechos de autor aunado al crecimiento de Internet. Y aunque las organizaciones han tenido éxito en sus esfuerzos por combatir la piratería de software, los gobiernos alrededor del mundo deben tomar los pasos que los lleven a mejorar sus leyes de propiedad intelectual y sistemas para el cumplimiento de las órdenes.

A continuación se presentan algunos datos estadísticos para tener una mayor visión de las pérdidas económicas que representa la piratería de software.

Los datos siguientes son resultado de un estudio realizado por la BSA (Business Software Alliance) que es la principal organización dedicada a promover un mundo seguro y legal. La BSA informa a los usuarios de computadoras sobre los derechos de autor del software y sobre la seguridad cibernética, apoya la política pública que fomenta la innovación e incrementa las oportunidades de negocio y lucha contra la piratería de software. Fue fundada en 1988, y cuenta con programas en 65 países de todo el mundo.

A comienzos del año 2002, la BSA realizó su séptimo análisis de la piratería de programas informáticos a nivel mundial correspondiente al año 2001. El objetivo del estudio consistió en analizar los datos disponibles y utilizar una metodología sistemática para determinar los índices mundiales de piratería de programas informáticos de uso empresarial y las pérdidas en dólares relacionadas con esta piratería. En este estudio, la piratería de programas informáticos se midió en base a la cantidad de programas informáticos de uso empresarial instalados o reproducidos durante el año 2001 sin su correspondiente licencia de uso. El impacto de la violación de los derechos de propiedad provocó pérdidas por casi \$10,967.309 millones de dólares en todo el mundo en el año 2001.

Los resultados del estudio realizado arrojaron los siguientes porcentajes de piratería de software:

Región	Índices de piratería						Ingresos perdidos en dólares (x 1000)					
	1996	1997	1998	1999	2000	2001	1996	1997	1998	1999	2000	2001
Europa Occidental	43%	39%	46%	34%	34%	37%	\$2,574,271	\$2,518,726	\$2,760,337	\$3,629,371	\$3,079,256	\$2,659,886
Europa Oriental	80%	77%	76%	70%	63%	67%	\$ 782,508	\$ 561,356	\$ 640,015	\$ 505,213	\$ 404,491	\$ 434,627
América del Norte	28%	28%	26%	26%	25%	26%	\$2,718,251	\$3,074,266	\$3,195,821	\$3,631,212	\$2,937,437	\$1,997,008
América Latina	69%	64%	62%	59%	58%	57%	\$ 980,568	\$ 977,994	\$1,045,506	\$1,127,639	\$ 869,777	\$ 864,700
Asia - Pacífico	55%	52%	49%	47%	51%	54%	\$3,739,304	\$3,916,236	\$2,954,812	\$2,791,531	\$4,083,061	\$4,726,454
Oriente Medio - África	74%	65%	63%	60%	55%	52%	\$ 510,756	\$ 391,510	\$ 380,019	\$ 478,192	\$ 376,344	\$ 284,634
TOTAL MUNDIAL	43%	40%	38%	36%	37%	40%	\$11,306,258	\$11,440,088	\$10,976,510	\$12,163,158	\$11,750,365	\$10,967,309

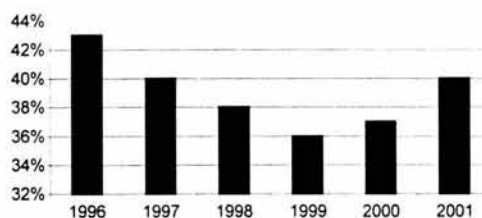


Figura I.2 Índices de piratería a nivel mundial

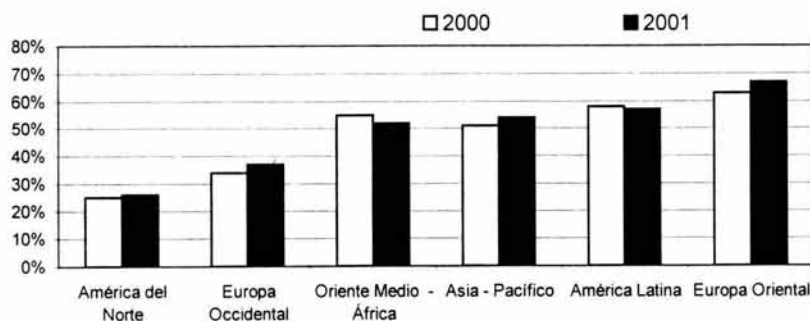


Figura I.3 Índices de piratería por región durante los años 2000 y 2001

En México: Las pérdidas por piratería en México fueron de 181.678 millones de dólares. En porcentaje corresponde al 55%, respectivamente.

Los resultados derivados del estudio mundial de piratería de BSA correspondiente al año 2001 indican que la piratería de programas informáticos

continúa siendo un desafío para la Industria Informática. Por primera vez en la historia del estudio, el índice mundial de piratería de programas informáticos aumentó durante dos años consecutivos, 2000 y 2001. El índice de piratería de 40% correspondiente al año 2001 muestra un marcado incremento con respecto a la cifra de 37% registrada en el año 2000. Ambos años muestran un aumento en comparación con el 36% registrado en 1999. Las pérdidas en dólares a causa de la piratería de programas informáticos registraron un descenso de 6,7% con respecto al año 2000 y se situaron en 10.970 millones de dólares. Este descenso en dólares no es indicador de un descenso en la tasa de piratería de programas informáticos. En realidad, es el resultado de otros factores. En el año 2001, el dólar estadounidense mantuvo una posición de fortaleza con respecto a otras monedas locales y los precios de los programas informáticos en dólares estadounidenses continuaron bajando, siguiendo una tendencia de descenso en los precios observada durante la última década.

El efecto de este descenso en los precios de los programas informáticos disimula, en parte el impacto de la pérdida en dólares causada por la piratería. Además, el mercado legal de programas informáticos registró una contracción en el año 2001, ya que los efectos de la recesión repercutieron en los gastos en materia de tecnología por parte de las empresas. La combinación de la contracción del mercado legal de programas informáticos sumado al descenso en los precios de programas informáticos tuvo como resultado una leve reducción en las pérdidas en dólares causadas por la piratería de programas informáticos. Pero al situarse en 10.970 millones de dólares, esta pérdida sólo se encuentra 10% por debajo de su pico máximo de 12.160 millones de dólares alcanzado en 1999. Las regiones que registraron pérdidas más elevadas en dólares en el año 2001 fueron Asia/Pacífico, Europa Occidental, y América del Norte. En estas regiones se encuentran las economías más desarrolladas y, por consiguiente, los mercados más grandes de computadoras personales y programas informáticos. En Europa Occidental y América del Norte, los índices relativamente bajos de piratería aún se traducen en grandes pérdidas en dólares.

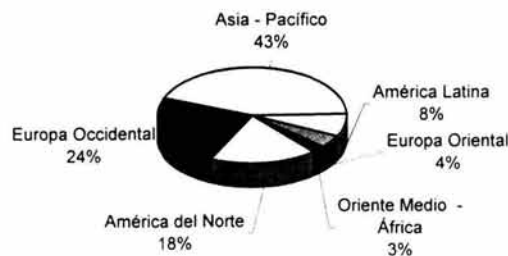


Figura I.4 Distribución de las pérdidas por región

Frente a todo lo analizado se puede determinar, que la piratería se ha convertido en una práctica muy extendida en el mundo actual, no solamente por particulares sino también por empresas y que la principal razón para que aumente cada vez más, es el argumento del ahorro en dinero, bajo el común fundamento y justificación de que la copia va a ser utilizada exclusivamente en el ámbito familiar o del trabajo y no se va a comercializar con ella.

Finalmente ante el supuesto ahorro, se pasan por alto riesgos como el de un daño grave a la computadora o pérdida irreversible de información, que además de los aspectos legales sancionadores a esta clase de reproducción puede salir costando muchísimo más que un programa original.

Como pudo apreciarse anteriormente la piratería es un problema mundial que crece día a día, es por ello que las empresas deben buscar la mejor manera de proteger sus programas, para esto el presente trabajo de tesis esta dedicado al desarrollo de un sistema integral que permita proteger el software de la piratería, pero antes de comenzar con la planeación y desarrollo del mismo, a continuación se describen y determinan algunos aspectos importantes como son el ciclo de vida y metodología a utilizar para el correcto desarrollo del mismo.

Todo proyecto de ingeniería tiene un fin ligado a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases, al conjunto de fases empleadas se le denomina "Ciclo de vida".

CICLO DE VIDA

Un modelo de ciclo de vida define el estado de las fases a través de las cuales se mueve un proyecto de desarrollo de software. La definición de un ciclo de vida facilita el control sobre los tiempos en que es necesario aplicar recursos de todo tipo al proyecto. Una fase es un conjunto de actividades relacionadas con un objetivo en el desarrollo del proyecto. Se construye agrupando tareas (actividades elementales) que pueden compartir un tramo determinado del tiempo de vida de un proyecto.

Cuanto más grande y complejo sea un proyecto, mayor detalle se necesita en la definición de las fases. De esta forma, cada fase de un proyecto puede considerarse un “micro-proyecto”.

Un modelo de ciclo de vida de software es un conjunto de actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas. Un modelo de ciclo de vida de software:

1. Describe las fases principales de desarrollo de software.
2. Define las fases primarias esperadas de ser ejecutadas durante esas fases.
3. Ayuda a administrar el progreso del desarrollo.
4. Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

Así, los modelos por una parte suministran una guía para los ingenieros de software con el fin de ordenar las diversas actividades técnicas en el proyecto, por otra parte suministran un marco para la administración del desarrollo, en el sentido en que permiten estimar recursos, definir puntos de control intermedios, monitorear el avance, etc.

El sistema a desarrollar en este trabajo de tesis empleará el modelo de ciclo de vida cascada, debido a que este modelo permite que el desarrollo del software sea a través de una secuencia simple de fases. Cada fase tiene un conjunto de metas y actividades bien definidas, lo que contribuye a alcanzar los objetivos.

Modelo de Ciclo de Vida Cascada

Este modelo divide el ciclo de vida del producto de programación en una serie de actividades sucesivas; cada fase requiere información de entrada, procesos y resultados, bien definidos.

En la siguiente figura se muestra el Modelo de ciclo de vida cascada.

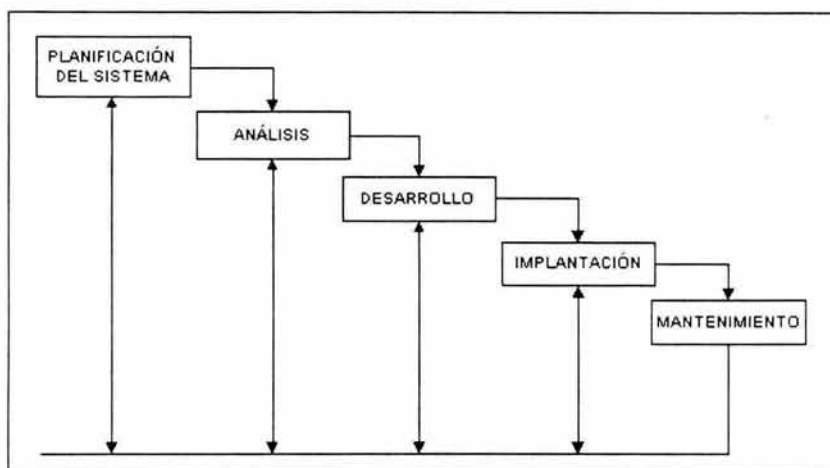


Figura I.5 Ciclo de vida cascada

A continuación se describen cada una de las fases de este modelo de ciclo de vida:

Planeación del sistema

Es la etapa en la que se determina si el proyecto es o no factible de realizar y se determinan tiempos y costos aproximados, estableciendo así la ruta crítica de cada actividad. Esto es porque la falta de planeación de un sistema es la causa principal de retrasos en programación, incremento de costos, poca calidad, y altos costos de mantenimiento en los desarrollos de productos de software.

Con frecuencia se dice que es imposible realizar una planeación inicial, porque la información precisa sobre las metas del proyecto, necesidades del cliente y restricciones del producto no se conocen al comenzar el proyecto de desarrollo, sin embargo, uno de los principales propósitos de esta fase es aclarar los objetivos, problemas o necesidades y restricciones.

Análisis

Es indispensable comprender perfectamente los requisitos del software, para que éste no fracase. Esta etapa puede parecer una tarea relativamente sencilla, pero las apariencias engañan. Abundan los casos en que se pueden llegar a malas interpretaciones o falta de información.

Desarrollo:

Dentro del desarrollo se contemplaran:

Diseño: El diseño del software es realmente un proceso de múltiples pasos que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

Codificación: El diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.

Implantación:

Esta fase involucra la instalación del software en el equipo de cómputo, una vez que se ha realizado la implantación comienza la etapa de prueba del programa

Prueba: La prueba se centra en la lógica interna del software, y en las funciones externas, asegurando que todas las sentencias se han evaluado, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requiere.

Mantenimiento

Es indudable que el software una vez entregado al cliente sufrirá cambios. Los cambios ocurrirán debido a que hayan encontrado errores o bien a que el software deba adaptarse a posibles cambios.

El análisis y diseño del sistema nos permite analizar la entrada, el flujo, almacenamiento y salida de la información, para evitar que durante el desarrollo del sistema haya retrocesos debido a una mala planeación, mala definición de los requerimientos de los usuarios y cambios constantes durante el desarrollo.

Metodología de desarrollo de software

En el desarrollo del sistema propuesto se combina el ciclo de vida junto una metodología estructurada ya que a lo largo de las fases de análisis y diseño se debe proceder paso a paso obteniendo retroalimentación de los usuarios y analizando el diseño buscando omisiones de errores. Esta metodología nos permite mejorar la operación del sistema al detectar problemas y oportunidades de mejoras.

Metodología Estructurada

Tradicionalmente, el análisis es visto como la definición de los requisitos del sistema. Por otro lado, el diseño establece una arquitectura para el software tal que satisfaga dichos requisitos y permita una posterior implantación ajustada a ciertos criterios de calidad. Dicho de otro modo el análisis detalla el *qué* mientras que el diseño desarrolla el *cómo*.

El diseño estructurado tuvo sus comienzos con la programación estructurada, que incorpora las tres estructuras de control básicas de un programa: secuencia, selección e iteración. El análisis estructurado aparece como complemento al diseño estructurado. Así, el propósito del Análisis Estructurado es proveer una técnica de análisis que permita una adecuada transición hacia el Diseño Estructurado. El análisis estructurado está basado en el modelado de flujos de información y su transformación mediante *procesos*, es decir, el analista se preocupa fundamentalmente de las funciones que debe llevar a cabo el sistema. Las propuestas iniciales del Análisis y Diseño Estructurado se centraban en el modelado de procesos dejando en segundo término el modelado de datos. El enfoque Estructurado Moderno sugiere realizar paralelamente el modelado de procesos y de datos (este último mediante Diagramas Entidad-Relación y modelado relacional de datos).

Análisis Estructurado

La metodología de análisis estructurado fue desarrollada por Edward Yourdon y Tom DeMarco en 1976 con el objeto de proveer un modelo centrado en el flujo de los datos de un sistema, en vez del enfoque del flujo de control, muy usado por los métodos tradicionales descendentes (Top Down). El objetivo fundamental del análisis estructurado es el de construir "especificaciones estructuradas", las cuales consisten de un documento donde se registran los requerimientos funcionales de un sistema. Las especificaciones estructuradas se diferencian de los requerimientos funcionales tradicionales en que estos últimos

son basados en texto, mientras las especificaciones estructuradas son basadas en gráficas.

El análisis estructurado se concentra en especificar lo que se requiere que haga el sistema o la aplicación. Permite que las personas observen los elementos lógicos (lo que hará el sistema) separados de los componentes físicos (computadora, terminales, sistemas de almacenamiento, etc). Después de esto se puede desarrollar un diseño físico eficiente para la situación donde será utilizado. No garantiza que las especificaciones sean correctas, pero puede acomodar más fácilmente los cambios y minimizar su impacto, ayudando a organizar los módulos de software en una forma funcional y fácil de mantener.

Análisis de Requerimientos

El análisis de requerimientos facilita al ingeniero de sistemas especificar la función y comportamiento de los programas, indicar la interfaz con otros elementos del sistema y establecer las ligaduras de diseño que debe cumplir el programa. El análisis de requerimientos permite al ingeniero refinar la asignación de software y representar el dominio de la información que será tratada por el programa. Proporciona al diseñador la representación de la información y las funciones que pueden ser traducidas en datos, arquitectura y diseño. Finalmente, la especificación de requerimientos suministra al programador los medios para valorar la calidad de los programas, una vez que se haya construido.

Visiones Lógicas y Físicas

La visión lógica de los requerimientos del software presenta las funciones que han de realizarse y la información que ha de procesarse independientemente de los detalles de implementación.

La visión física de los requerimientos del software presenta una manifestación del mundo real de las funciones de procesamiento y las estructuras de información. En algunos casos se desarrolla una representación física como el primer paso del diseño del software. Sin embargo la mayoría de los sistemas basados en computador, se especifican de forma que se dictan ciertas recomendaciones físicas.

Una vez que se hayan descrito las funcionalidades básicas, comportamiento, interfase e información, se especifican los criterios de validación para demostrar una comprensión de una correcta implementación de los programas. Estos criterios sirven como base para hacer una prueba durante el desarrollo de los programas. Para definir las características y atributos del software se escribe una especificación de requerimientos formal.

Partición (Modularidad)

Normalmente los problemas son demasiado grandes y complejos para ser comprendidos como un todo. Por esta razón, se tiende a particionar (dividir) tales problemas en partes que puedan ser fácilmente comprendidas, y establecer interfases entre las partes, de forma que se realice la función global. Durante el análisis de requerimientos, el dominio funcional y el dominio de la información del software pueden ser particionados. En esencia la partición descompone un problema en sus partes constituyentes. Conceptualmente, se establece una representación jerárquica de la función o información y luego se parte el elemento superior mediante: 1) incrementando los detalles, moviéndonos verticalmente en la jerarquía, 2) descomponiendo funcionalmente el problema, moviéndonos horizontalmente en la jerarquía.

Diagramas de Flujos de Datos

Conforme la información se mueve a través del software, se modifica mediante una serie de transformaciones. Un diagrama de flujos de datos (DFD), es una técnica grafica que describe el flujo de información y las transformaciones que se aplican a los datos, conforme se mueven de la entrada a la salida.

Diagramas de flujos de datos Lógicos y Físicos:

- **Diagrama lógico:** Se enfoca en el negocio y la manera en que opera el mismo, describe los eventos que suceden y los datos requeridos y producidos por cada evento. Se centra en el flujo de datos entre los procesos, sin considerar los dispositivos específicos y la localización de los almacenes de datos o personas en el sistema. Sin indicarse las características físicas.
- **Diagrama físico:** Muestra cómo será implantado el sistema, incluyendo el hardware, software, archivos y personas involucradas en el sistema.

Diccionario de Datos

El diccionario de datos contiene las definiciones de todos los datos mencionados en el DFD, en una especificación del proceso y en el propio diccionario de datos. Los datos compuestos (datos que pueden ser además divididos) se definen en términos de sus componentes; los datos elementales (datos que no pueden ser divididos) se definen en términos del significado de cada uno de los valores que puede asumir. Por tanto, el diccionario de datos esta compuesto de definiciones de flujo de datos, archivos [datos almacenados] y datos usados en los procesos [transformaciones].

Requerimientos de las Bases de Datos

El análisis de requerimientos para una base de datos incorpora las mismas tareas que el análisis de requerimientos del software. Es necesario un contacto estrecho con el cliente; es esencial la identificación de las funciones e interfases; se requiere la especificación del flujo, estructura y asociatividad de la información y debe desarrollarse un documento formal de los requerimientos.

Diseño Estructurado

El objetivo del Diseño Estructurado es el generar programas formados por módulos independientes unos de otros desde el punto de vista funcional. La metodología de diseño estructurado ha ido evolucionando desde que surge a mediados de los años 70s, ha despertado interés y ha sido utilizada mayoritariamente desde entonces. Se han introducido mejoras por varios autores entre los cuales se pueden mencionar: J. D. Warnier, M.A. Jackson, y J.G. Myers.

El diseño de sistemas estructurados en datos (DSED, ó DSSD, por sus siglas en inglés), también denominado metodología de Warnier-Orr, ha evolucionado del innovador trabajo sobre el análisis del campo de información realizado por J. D. Warnier.

Kenneth Orr amplió el trabajo de Warnier para abarcar una visión algo más amplia del campo de información, evolucionando el desarrollo de sistemas estructurados de datos. El DSED considera el flujo de la información y las características funcionales, así como la jerarquía de los datos.

El Desarrollo de Sistemas Jackson (DSJ) evolucionó a partir del trabajo de M.A. Jackson sobre análisis del campo de la información y sus relaciones con el diseño de programas y de sistemas. Similar, de alguna forma al enfoque de Warnier y el DSED, el DSJ se centra en modelos del ámbito de información del "mundo real". En palabras de Jackson "El desarrollador del software comienza creando un modelo de la realidad concerniente al sistema, la realidad que comprende su materia base".

Cuando un sistema ha sido descrito usando diagramas de flujo de datos y va a ser implementado como un conjunto de procesos paralelos, estas organizaciones lógicas deben convertirse en estructuras físicas. Sin embargo, antes de llegar a las estructuras físicas el software debe ser organizado en una jerarquía de componentes, donde las transformaciones de datos sean agrupables bajo un único padre.

Los cuadros estructurados son el principio fundamental del diseño estructurado y describen un sistema como una jerarquía de partes, mostrándolas

gráficamente como un árbol. Estas estructuras documentan la transformación de los elementos de un diagrama de flujo de datos en una jerarquía de módulos de programas. El propósito del cuadro estructurado es describir la organización del software mostrando como los componentes están formados de subcomponentes ó módulos. En resumen, el diseño estructurado es una metodología gráfica para ilustrar las interconexiones entre módulos de software o alguna otra información, tales como secuencia de ejecución de módulos y sus interacciones.

La metodología contempla algunos principios básicos de diseño, como son:

- Simplicidad
- Definición exacta de entradas y salidas a los módulos.
- Localidad (Un módulo debe contener el código y los datos necesarios para llevar a cabo su función y sólo debe invocar otros módulos que estén bajo su estructura)
- La ejecución de un módulo no debe afectar la operación de otro módulo.

El Modelo de Datos

El modelo de datos es el proceso de identificación de los elementos de datos, las estructuras de los datos, los formatos y las relaciones entre las entidades del sistema o aplicación. El modelo de datos ha dado origen en los últimos años al concepto de Modelado de Información; en los últimos 15 años han evolucionado varias metodologías para modelación de datos e información, pero las dos más usadas son los diagramas entidad-relación y los diagramas de Warnier-Orr. Los diagramas entidad-relación fueron inventados por Peter Chen en 1976 y son usados para describir relaciones entre datos. Los diagramas de Warnier-Orr representan una técnica simple y sencilla para presentar estructuras de sistemas y pueden ser usados como una herramienta para modelar datos o como una herramienta para modelar estructuras de software.

El diseño es cómo armar cada uno de los componentes de un sistema, que es un conjunto de elementos interrelacionados, con factores internos y externos que afectan a una empresa y donde siempre hay una entrada, un proceso que incluyen las bases de datos y una salida. También se diseña la interfaz del usuario, que es el cómo aprovechar el hardware a través del software.

Diseño Lógico

El diseño debe comenzarse por la salida, ya que es la información que se entrega a los usuarios por medio de un sistema de información. Puede tomar muchas formas, la permanente y tradicional de los reportes impresos y la fugaz de pantallas, micro formas y sonido. Los usuarios dependen de la salida para realizar las tareas y juzgan el mérito de un sistema únicamente por su salida.

Objetivos que debe cumplir la salida:

- 1) Diseñar la salida para que sirva al propósito deseado, debe ser una solución a los requerimientos y objetivos del proyecto, revelados durante la etapa de análisis y determinación de los requerimientos de información.
- 2) Diseñar la salida para que se ajuste al usuario, en muchos casos debe presentarse en forma multidimensional, con base en técnicas de revelamiento (entrevistas, observación, etc) se diseña el sistema atendiendo a lo que los usuarios necesitan.
- 3) Entregar la cantidad y calidad adecuada de salida, que debe dimensionarse en función al costo-beneficio. A nadie le sirve información en exceso, sólo para mostrar la capacidad del sistema.
- 4) Asegurarse de que la salida se encuentre donde se necesita, la distribución adecuada es importante para su uso y utilidad, debe ser presentada al usuario adecuado.
- 5) Entrega de la salida a tiempo, una de las quejas más comunes de los usuarios es que no reciben la información a tiempo para tomar decisiones, la temporización adecuada de la salida puede ser crítica para las operaciones de negocios.
- 6) Selección del método de salida adecuado con el dirigido hacia sistemas en línea, mucha información ahora tiene la salida por pantalla exclusivamente. Los costos, la flexibilidad, el tiempo de vida, distribución, almacenamiento, posibilidades de recuperación, transportabilidad e impacto sobre los datos y la velocidad difieren entre los distintos métodos.

Selección de la tecnología de salida:

a) **Impresoras:** Hay tres factores principales de la impresora a tener en cuenta, *la confiabilidad:* hace lo que se espera, durante una vida útil determinada, *la compatibilidad con software y hardware:* importante cuando los sistemas son modificados o se ponen en red con otros sistemas más pequeños y *el soporte del fabricante:* es muy importante el servicio post-venta del fabricante, como el reemplazo de la maquina mientras está siendo reparada, o el servicio técnico en casa del cliente.

b) Pantallas: son la tecnología cada vez más popular, las ventajas son su silenciosidad, su capacidad de participación interactiva, es ideal para brindar información que se necesita una sola vez y no necesita ser guardada.

c) Salida de audio: es opuesto a la impresora, es transitoria, lo impreso es permanente, es para el usuario, reloj parlante, confirmación de lo teclado como la calculadora parlante o sobre el borrado de archivos, comentario parlante.

d) Microformas: microfichas o microfilm, son formas de almacenar grandes volúmenes de información (la centésima parte), la desventaja es la necesidad de adquirir y mantener nuevo equipo y capacitar al usuario sobre su uso.

e) CD-Rom: se usan para la salida de información voluminosa de contenido estable. La recuperación es mucho más rápida que otros métodos. Pueden incluir textos, gráficos, música y vídeo.

f) Electrónica: muchos de los sistemas tienen la capacidad de enviar información por medio de correo electrónico (interno de la organización o por servicios de comunicación para intercambio externo), fax, mensajes de tablero electrónico y por computadora sin necesidad de papel.

La calidad de la entrada de un sistema determina la calidad de la salida. Es vital que las formas y pantallas de entrada sean diseñadas pensando en esta crítica relación.

Los lineamientos para las formas de entrada bien diseñadas son: satisfacer los objetivos de efectividad, que sirva a propósitos específicos de manejo de información, de precisión: se refiere al diseño que asegura el llenado adecuado, de facilidad de uso: que las formas y pantallas sean concisas y no requieran tiempo adicional para su entendimiento. Las formas son instrumentos importantes para dirigir el curso del trabajo: Son papeles estándares impresos llenados por la gente, que extraen y capturan información requerida por miembros de la organización.

Los cuatro lineamientos para pantallas bien diseñadas son: ser fáciles de llenar, atractivas, precisas y que satisfagan los objetivos para los que fueron diseñadas. Por otro lado se debe realizar un diseño de las entradas, para ello se cuenta con el diccionario de datos. La entrada debe condicionar los datos incompletos, incorrectos o no autorizados. El sistema debe discriminar los datos que son imprescindibles de los que no lo son y pueden omitirse, debe establecer donde deben efectuarse los controles, si se hacen en una terminal no puede ser una terminal tonta. Además hay que prever que algunos datos pueden ser tomados de formularios, por ejemplo, cupones de venta de tarjeta de crédito, en estos casos,

también debe diseñarse el formulario, que deben conducir a la agilidad del trabajo del operador.

Debe diseñarse la codificación para el ingreso de datos, tratando de que los mismos sean representativos de los datos que se desean ingresar, las formas de ingreso son por medio del teclado, lápiz óptico, lectoras de código de barras, lectura óptica, lectura de caracteres en tinta magnética (cheques), lectura de marcas sensibles, etc.

Interfaz de Usuario

La interfaz de usuario es la parte más importante y crítica de una aplicación. Es la única parte del sistema que el usuario puede ver. Si esta interfaz llega a ser torpe, lenta o difícil de aprender, no importa que tan bien este diseñado el software internamente, el usuario final juzgará la aplicación como de baja calidad.

Las interfases gráficas con el usuario han surgido en los últimos 10 años debido a la gran diseminación de los sistemas interactivos y a las facilidades de graficación que presentan las nuevas generaciones de lenguajes de programación. El diseño de la interfaz de usuario generalmente comienza con la definición de un prototipo con el cual el usuario puede experimentar parte del futuro comportamiento del sistema. La principal función del prototipo es la de establecer los requerimientos de sistema.

Los objetivos de diseñar interfases para ayudar a los usuarios a proporcionar información que necesitan son: la efectividad para acceder al sistema, aumento de la velocidad en la captura de datos, reducción de errores, logro de retroalimentación del sistema a los usuarios y la productividad.

Tipos de Interfaz: Tiene dos componentes principales, el lenguaje de presentación (de la computadora al usuario) y el lenguaje de acción (la parte del usuario a la computadora)

- **Interfases de lenguaje natural:** permite que los usuarios interactúen en su lenguaje con la computadora. No se requieren habilidades especiales del usuario. Los problemas de implementación son mínimos.
- **Interfases de pregunta y respuesta:** la computadora muestra una pregunta, el usuario teclea una respuesta y la computadora actúa sobre esa información en forma preprogramada, moviéndose a la siguiente pregunta.
- **Interfases de lenguaje comando:** permite al usuario controlar la aplicación con una serie de tecleos, comandos, frases o secuencia de ellos. No tiene significado inherente al usuario, requieren memorización de las reglas y

puede ser un obstáculo, lo prefieren los usuarios experimentados por la rapidez.

- **Interfases gráficas de usuario:** permiten el manejo de la representación gráfica en pantalla, requiere mas sofisticación del sistema, algunas interfases gráficas utilizan iconos, como Windows.
- **Otras interfases de usuario:** incluyen dispositivos como plumas ópticas, pantallas sensibles al tacto, reconocimiento de voz. La ventaja es la aceleración en la captura de datos.

Para la evaluación de la interfase seleccionada hay que tener en cuenta: que el período de capacitación para los usuarios debe ser corto, el tiempo para el regreso de los errores del sistema debe ser corto y los usuarios poco frecuentes deben ser capaces de aprender rápidamente.

Retroalimentación para usuarios:

Es necesaria la retroalimentación a los usuarios por parte del sistema, para que sepan si su entrada está siendo aceptada, con datos correctos, si el procesamiento está avanzando, si las peticiones pueden ser o no procesadas y si se encuentra disponible información más detallada y cómo obtenerla. También puede ser efectiva la retroalimentación por audio.

Programación estructurada

El creciente empleo de las computadoras ha conducido a buscar un abaratamiento del desarrollo de software, paralelo a la reducción del costo del hardware obtenido gracias a los avances tecnológicos. Los altos costos del mantenimiento de las aplicaciones en producción normal también han surgido de la necesidad de mejorar la productividad del personal de programación.

En la década de los sesentas salieron a la luz pública los principios de lo que más tarde se llamo Programación Estructurada, posteriormente se libero el conjunto de las llamadas "Técnicas para mejoramiento de la productividad en programación" (en ingles Improved Programming Technologies, abreviado IPTs), siendo la Programación Estructurada una de ellas.

Los programas computarizados pueden ser escritos con un alto grado de estructuración, lo cual les permite ser fácilmente comprensibles en actividades tales como pruebas, mantenimiento y modificación de los mismos. Mediante la programación Estructurada todas las bifurcaciones de control de un programa se encuentran estandarizadas, de forma tal que es posible leer la codificación del mismo desde su inicio hasta su terminación en forma continua, sin tener que saltar

de un lugar a otro del programa siguiendo el rastro de la lógica establecida por el programador, como es la situación habitual con codificaciones desarrolladas bajo otras técnicas.

Las pruebas de los programas desarrollados utilizando este método, se acoplan mas rápidamente y el resultado final son programas que pueden ser leídos, mantenidos y modificados por otros programadores con mayor facilidad.

La programación estructurada es una técnica en la cual la estructura de un programa, se realiza mediante el uso de tres estructuras lógicas de control:

- a. Secuencia: Sucesión simple de dos o más operaciones.
- b. Selección: bifurcación condicional de una o más operaciones.
- c. Iteración: Repetición de una operación mientras se cumple una condición.

Estos tres tipos de estructuras lógicas de control pueden ser combinados para producir programas que manejen cualquier tarea de procesamiento de información. Un programa estructurado esta compuesto de segmentos, los cuales pueden estar constituidos por pocas instrucciones o por una página o más de codificación. Cada segmento tiene solamente una entrada y una salida, estos segmentos, asumiendo que no poseen lazos infinitos y no tienen instrucciones que jamás se ejecuten, se denominan programas propios. Cuando varios programas propios se combinan utilizando las tres estructuras básicas de control mencionadas anteriormente, el resultado es también un programa propio.

La programación estructurada esta basada en el teorema de la estructura, el cual establece que cualquier programa propio es equivalente a un programa que contiene solamente las estructuras lógicas mencionadas anteriormente. Una característica importante en un programa estructurado es que puede ser leído en secuencia, desde el comienzo hasta el final sin perder la continuidad de la tarea que cumple el programa, lo contrario de lo que ocurre con otros estilos de programación y en ejecución no tiene partes por las cuales nunca pasa ni tiene ciclos infinitos. Esto es importante debido a que, es mucho más fácil comprender completamente el trabajo que realiza una función determinada, si todas las instrucciones que influyen en su acción están físicamente contiguas y encerradas por un bloque. La facilidad de lectura, de comienzo a fin, es una consecuencia de utilizar solamente tres estructuras de control y de eliminar la instrucción de desvío de flujo de control, excepto en circunstancias muy especiales tales como la simulación de una estructura lógica de control en un lenguaje de programación que no la posea.

ESTRUCTURAS DE CONTROL

Las estructuras de control de un lenguaje de programación se refieren a el orden en que las instrucciones de un algoritmo se ejecutarán. El orden de ejecución

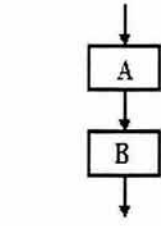
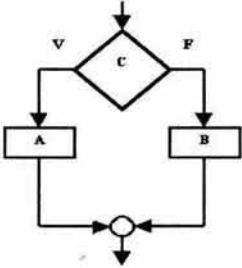
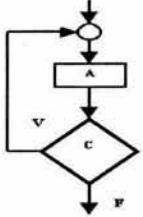
de las sentencias o instrucciones determinan el control. Estas estructuras de control son por consiguiente fundamentales en los lenguajes de programación y en los diseños de algoritmos, especialmente los pseudocódigos.

La programación estructurada hace los programas más fáciles de escribir, verificar, leer y mantener, además de que utilizan un número limitado de estructuras de control que minimizan la complejidad de los problemas.

En síntesis la programación estructurada significa:

- El programa completo tiene un diseño modular
- Los módulos se diseñan con metodología descendente (método top-down)
- Cada módulo se codifica utilizando las 3 estructuras de control básicas: secuenciales, selectivas y repetitivas.
- Estructuración y modularidad que son complementarios mas no iguales.

Las tres estructuras de control básico son: secuencia, selección, repetición

<p>SECUENCIA</p> <p>Indica que las instrucciones de un programa se ejecutan una después de la otra, en el mismo orden que aparecen en el programa. Se representa gráficamente como una caja después de otra, ambas con una sola entrada y una única salida.</p>	
<p>SELECCIÓN</p> <p>También conocida como la estructura SI-CIERTO-FALSO, plantea la selección entre dos alternativas con base en el resultado de la evaluación de una condición o predicado; equivale a la instrucción IF de todos los lenguajes de programación.</p>	
<p>ITERACIÓN</p> <p>También llamada la estructura HACER-MIENTRAS-QUE, corresponde a la ejecución repetida de una instrucción mientras que se cumple una determinada condición.</p>	

Finalmente se puede concluir que con la programación estructurada elaborar programas para computadora sigue siendo una labor que demanda esfuerzo, creatividad, habilidad y cuidado. Sin embargo, con este estilo se obtiene las siguientes ventajas:

1. Los programas son más fáciles de entender. Un programa estructurado puede ser leído en secuencia, de arriba hacia abajo, sin necesidad de estar saltando de un sitio a otro en la lógica, lo cual es típico de otros estilos de programación. La estructura del programa es más clara puesto que las instrucciones están más ligadas o relacionadas entre sí, por lo que es más fácil comprender lo que hace cada función.
2. Reducción del esfuerzo en las pruebas. El programa se puede tener listo para producción normal en un tiempo menor del tradicional; por otro lado, el seguimiento de las fallas o depuración (debugging) se facilita debido a la lógica más visible, de tal forma que los errores se pueden detectar y corregir más fácilmente.
3. Reducción de los costos de mantenimiento.
4. Programas más sencillos y más rápidos.
5. Aumento en la productividad del programador.
6. Se facilita la utilización de las otras técnicas para el mejoramiento de la productividad en programación.
7. Los programas quedan mejor documentados internamente.

Hasta esta parte se ha descrito teóricamente el ciclo de vida en cascada que sigue el sistema propuesto por el presente proyecto de tesis, la metodología estructurada, la cual es empleada para la etapa de análisis del sistema y como parte de la misma metodología, se definió la programación estructurada para la fase de desarrollo del sistema.

Por otro lado es importante mencionar que la información que se maneja en el sistema integral propuesto, se almacena en una Base de Datos Relacional, para comprender mejor este concepto, a continuación se define lo que es una de base de datos, tipos, características y diseño.

Base de Datos

Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada. Un archivo por sí mismo no constituye una base de datos, sino más bien la forma en que está organizada la información es la que da origen a la base de datos.

Una base de datos contiene archivos que contienen registros que contienen campos



Figura I.6 Componentes de base de datos

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

Ventajas en el uso de bases de datos:

1. Globalización de la información: permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
2. Eliminación de información inconsistente: si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deben hacerse a todas las copias del archivo.
3. Permite compartir información.
4. Permite mantener la integridad en la información: la integridad de la información es una de las cualidades altamente deseable y tiene por objetivo que sólo se almacene la información correcta.
5. Independencia de datos: el concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos. La independencia de datos implica un divorcio entre programas y datos.

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

Sistema de Gestión de Bases de Datos (SGBD). Es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una BD nunca se accede o manipula directamente sino a través del SGBD. Se puede considerar el SGBD como la interfaz entre el usuario y la Base de datos.

Funciones de un SGBD

Un SGBD debe proporcionar un amplio surtido de funcionalidades para poder cumplir adecuadamente su misión. Normalmente se clasifican en definición, manipulación y utilización.

- **Función de definición:** Permite describir los elementos de datos, sus estructuras, sus interrelaciones y sus validaciones a nivel externo, lógico e interno. Esta función es realizada por una parte del SGBD denominada lenguaje de definición de datos (LDD o DDL, *Data Definition Language*).
- **Función de manipulación:** Permite buscar, añadir, suprimir y modificar los datos de la BD. Esta función es realizada por una parte del SGBD denominada lenguaje de manipulación de datos (LMD o DML, *Data Manipulation Language*).
- **Función de utilización:** Incluye otras funcionalidades tales como: modificar la capacidad de los registros, cargar archivos, realizar copias de seguridad, protección frente a accesos no autorizados, gestión de la concurrencia, estadísticas de utilización, etc.

El lenguaje de definición de datos

Permite definir un esquema de base de datos por medio de una serie de definiciones que se expresan en un lenguaje especial, el resultado de estas definiciones se almacena en un archivo especial llamado diccionario de datos.

Diccionario de datos: Es un archivo que contiene metadatos ³ que se consulta antes de leer o modificar datos reales en el sistema de base de datos. Los metadatos deben precisar información que nos indique con claridad el tipo de datos que son utilizados, sus ámbitos de influencia y sus limitantes de integridad.

³ **Metadato:** Es un dato que describe datos, contendrá el número de columnas de la tabla, el tipo de datos de estas columnas, cuáles de estas columnas pueden tener valores nulos, etc. Estos son datos sobre datos.

Lenguaje de manipulación de datos

La manipulación de datos se refiere a las operaciones de insertar, recuperar, eliminar o modificar datos; dichas operaciones son realizadas a través del lenguaje de manipulación de datos (DML, Data Manipulation Language), que es quien permite el acceso de los usuarios a los datos. Existen básicamente 2 tipos de lenguajes de manipulación de datos:

- **Procedimentales:**

Los DML requieren que el usuario especifique que datos se necesitan y cómo obtenerlos.

- **No procedimentales:**

Los DML requieren que el usuario especifique que datos se necesitan y sin especificar cómo obtenerlos.

Arquitectura de los sistemas de bases de datos

Hay tres características importantes de los sistemas de bases de datos: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un archivo para almacenar el esquema de la base de datos. Los sistemas de base de datos manejan una arquitectura de tres niveles cuyo objetivo es separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

1. En el *nivel interno* se describe la estructura física de la base de datos mediante un *esquema interno*. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
2. En el *nivel conceptual* se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un *esquema conceptual*. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo de datos conceptual o un modelo de datos lógico para especificar el esquema.
3. En el *nivel externo* se describen varios *esquemas externos* o *vistas de usuario*. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo de datos conceptual o un modelo de datos lógico para especificar los esquemas.

La arquitectura de tres niveles es útil para explicar el concepto de *independencia de datos* que se puede definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior.

Se pueden definir dos tipos de independencia de datos:

- **Independencia física de datos:** Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación.
- **Independencia lógica de datos:** Capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación.

Una de las características fundamentales de los sistemas de bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar las características sobre el almacenamiento físico que la mayoría de los usuarios no necesita conocer. Los modelos de datos son el instrumento principal para ofrecer dicha abstracción.

Un *modelo de datos* es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos se pueden clasificar dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos. Los modelos de datos de alto nivel, o *modelos conceptuales*, disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios percibe los datos, mientras que los modelos de datos de bajo nivel, o *modelos físicos*, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en la computadora. Los conceptos de los modelos físicos están dirigidos al personal informático, no a los usuarios finales. Entre estos dos extremos se encuentran los *modelos lógicos*, cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente.

Los modelos lógicos ocultan algunos detalles de cómo se almacenan los datos, pero pueden implantarse de manera directa en una computadora.

Los modelos conceptuales utilizan conceptos como entidades, atributos y relaciones. Una *entidad* representa un objeto o concepto del mundo real como, por ejemplo, un empleado de la empresa inmobiliaria o una oficina. Un *atributo* representa alguna propiedad de interés de una entidad como, por ejemplo, el nombre o el salario del empleado. Una *relación* describe una interacción entre dos o más entidades, por ejemplo, la relación de trabajo entre un empleado y su oficina.

Cada sistema de Base de Datos soporta un modelo lógico, siendo los más comunes el de red, el jerárquico y el relacional. Estos modelos representan los datos valiéndose de estructuras de registros, por lo que también se denominan

modelos orientados a registros. Hay una nueva familia de modelos lógicos, son los *modelos orientados a objetos*, que están más próximos a los modelos conceptuales.

Modelos lógicos basados en registros: Sus características principales son que permiten una descripción más amplia de la implantación, pero no son capaces de especificar con claridad las limitantes de los datos. Son ejemplos de este tipo de modelos:

Modelo de red: Los datos se representan mediante nombres de registros y las relaciones mediante conjunto de ligas. Es una variación del modelo de datos jerárquico. De hecho las bases de datos pueden traducirse de jerárquicas a en redes y viceversa con el objeto de optimizar la velocidad y la conveniencia del procesamiento. Mientras que las estructuras jerárquicas describen relaciones de muchos a muchos.

Modelo jerárquico: Es semejante al modelo de red, pero con una estructura arbolada. Una clase de modelo lógico de bases de datos que tiene una estructura arborescente. Un registro subdivide en segmentos que se interconectan en relaciones padre e hijo y muchos más. Los primeros sistemas administradores de bases de datos eran jerárquicos. Puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.

Modelo relacional: Los datos y las relaciones se representan mediante tablas, cada una con diferentes columnas y nombres únicos. Es el más reciente de estos modelos, supera algunas de las limitaciones de los otros dos anteriores. El modelo relacional de datos representa todos los datos en la base de datos como sencillas tablas de dos dimensiones llamadas relaciones. Las tablas son semejantes a los archivos planos, pero la información en más de un archivo puede ser fácilmente extraída y combinada. Este modelo es descrito más detalladamente a continuación ya que es el modelo a utilizar en este proyecto de tesis.

Modelo Entidad-Relación

Una de las características de este modelo es que permite representar con claridad las limitantes de los datos. El modelo E-R se basa en una percepción del mundo real, la cual esta formada por objetos básicos llamados entidades y las relaciones entre estos objetos así como las características de estos objetos llamados atributos.

Sus características principales son:

- Reflejan tan sólo la existencia de los datos sin expresar lo que se hace con ellos.
- Es independiente de las bases de datos y de los sistemas operativos (por lo que puede ser implementado en cualquier base de datos).
- Está abierto a la evolución del sistema.

- Incluye todos los datos que se estudian sin tener en cuenta las aplicaciones que se van a tratar.
- No tienen en cuenta las restricciones de espacio y almacenamiento del sistema.

Una *entidad* es un objeto que existe y se distingue de otros objetos de acuerdo a sus características llamadas atributos . Las entidades pueden ser concretas como una persona o abstractas como una fecha.

Un *conjunto de entidades* es un grupo de entidades del mismo tipo. Por ejemplo el conjunto de entidades CUENTA, podría representar al conjunto de cuentas de un banco X, o ALUMNO representa a un conjunto de entidades de todos los alumnos que existen en una institución.

Una entidad se caracteriza y distingue de otra por los *atributos*, en ocasiones llamadas propiedades, que representan las características de una entidad. Los atributos de una entidad pueden tomar un conjunto de valores permitidos al que se le conoce como **dominio** del atributo.

Una *tabla* es la forma de estructurar los datos en filas o registros y columnas o atributos.

Operaciones sobre Tablas

Todas las operaciones que podamos realizar sobre las tablas, vistas o elementos de ellas, están integradas en el SGDBR (Sistema Gestor de Bases de Datos Relacional) como rutinas. Ejemplos de operaciones son:

- **Selección:** Obtiene un subconjunto de filas de la tabla o vista, que cumplen una determinada condición.
- **Proyección:** Obtiene un subconjunto de columnas de todas las filas de la tabla.
- **Unión:** Realizamos la unión de varias tablas, cuyo resultado será el conjunto de todas las filas de las tablas origen. Las columnas respectivas de dichas tablas deben ser iguales entre sí.
- **Diferencia:** Inversa a la anterior, devuelve las filas que estén en una tabla y no pertenezcan a una segunda tabla. Deben por tanto ser iguales también las columnas respectivas entre sí.
- **Producto cartesiano:** El resultado será una fila por cada combinación entre cada fila de una tabla y todas las de la otra. Los valores de ambas filas se concatenarán.
- **Intersección:** Obtiene aquellas filas que sean idénticas en ambas tablas.

- **Join:** Es la operación de unir filas de dos tablas a través de algún campo común, dando como resultado filas con la suma de columnas de ambas tablas cuando se cumpla la condición del Join a través del campo (o campos) relacionados.

Relaciones y Conjuntos de Relaciones

Una relación es una asociación entre varias entidades.

La mayoría de las relaciones son Binarias; no obstante, pueden existir relaciones que incluyan a más de dos conjuntos de entidades.

Normalmente asocian a dos conjuntos de entidades y la relación tendrá una función determinada; a esta se le denomina papel. Normalmente se utilizan los papeles para etiquetar y así reconocer las relaciones establecidas. Las relaciones también pueden tener atributos descriptivos, en cuyo caso, la relación se describe indicando la pareja (atributo, último valor del atributo) sobre la relación. El modelo E-R permite definir una serie de limitantes aplicables en la información contenida en la base de datos, básicamente pueden definirse dos tipos de limitantes:

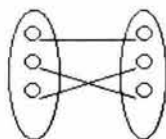
a) **CARDINALIDAD DEL MAPEO.-** es aquella mediante la cual puede especificarse la cantidad de entidades que podrán asociarse mediante una relación. La cardinalidad del mapeo se aplica generalmente sobre dos conjuntos de entidades. Las cardinalidades existentes para dos conjuntos de entidades A y B y conjunto de relaciones R pueden ser:

1. **Una a una:** Una entidad de A puede asociarse únicamente con una entidad de B.
2. **Una a muchas:** Una entidad de A puede asociarse con cualquier cantidad de entidades de B.
3. **Muchas a una:** Cualquier cantidad de entidades de A puede asociarse con una entidad de B.
4. **Muchas a muchas:** Cualquier cantidad de entidades de A puede asociarse con cualquier cantidad de entidades en B.

Ejemplo:

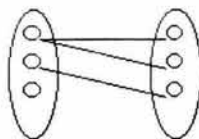
UNA A UNA
Alumnos Tesis

A B



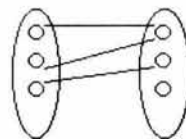
UNA A MUCHAS
Carrera Alumnos

A B



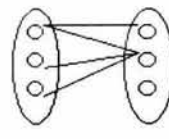
MUCHAS A UNA
Alumnos Carrera

A B



MUCHAS A MUCHAS
Alumnos Materias

A B



- b) **DEPENDENCIA DE EXISTENCIA.**- Nos permiten definir que un conjunto de entidades esta condicionado a la existencia de otro, un ejemplo de este condicionamiento se da entre una entidad alumno y la entidad calificación. A esta limitante se le denomina dependencia por existencia. Si una entidad Y requiere de una entidad X para existir se dice que Y es dependiente por existencia de X; esto implica que si eliminamos a la entidad X; deberá eliminarse la entidad Y. Para el caso anterior, se nombrara a X como la entidad dominante, y a Y como entidad subordinada.

Llaves primarias

Uno de los procesos de mayor relevancia en la manipulación de una base de datos es el de distinguir entre las diversas entidades y relaciones que son manipuladas. Se entiende como una llave al medio que nos permite identificar en forma unívoca (única e inequívoca) a una entidad dentro de un conjunto de entidades. Existen diversas categorías que permiten clasificar los tipos de llaves a utilizar:

1. **SUPER -LLAVE .-** Es un conjunto de atributos mediante los cuales es posible reconocer a una entidad. Este tipo de llaves contiene comúnmente atributos ajenos; es decir; atributos que no son indispensables para llevar a cabo el reconocimiento del registro.

Ejemplo:

Conjunto de entidades: Cursos

Atributos Súper llaves

*Nombre materia Nombre, mat, carrera, semestre

*Carrera Nombre, mat, carrera, unidades

*Semestre Nombre, mat, carrera, semestre, periodo

*Periodo Nombre, mat, carrera

*Unidades

-Si el conjunto de atributos X es una súper llave entonces cualquier conjunto de X será súper-llave.

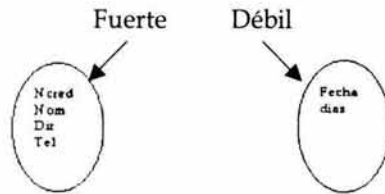
2. **LLAVE CANDIDATO.**- Son aquellas súper llaves que no contienen atributos ajenos; es decir, aquellos conjuntos de atributos que no tienen un subconjunto menor que pueda considerarse como súper llave.

3. **LLAVE PRIMARIA.**- Es aquella llave que el diseñador de la base de datos selecciona entre las llaves candidatos encontradas.

Existen conjuntos de entidades que no poseen los atributos necesarios para conformar una llave primaria; se les conoce como entidad débil. Cuando

existen los atributos necesarios para formar una llave primaria, se denominan entidad fuerte. Las entidades débiles se subordinan a las entidades fuertes.

Ejemplo:



Esquema de integridad

Un control de integridad o restricciones es aquel que nos permite definir con precisión el rango de valores validos para un elemento y/o las operaciones que serán consideraciones validas en la relación de tales elementos.

El objetivo primordial de un control de integridad es la reducción de la inconsistencia en la BD. Las restricciones de integridad normalmente se aplican en tres niveles:

1. UN ATRIBUTO SIMPLE.- Se define un dominio del atributo que es totalmente independiente del resto del entorno de la Base de Datos.
2. UN ATRIBUTO DEPENDIENTE DE OTRO.- Se definen subconjuntos de dominios posibles para un atributo X según el valor que previamente a sido asignado al atributo W.
3. RELACIONES ENTRE TUPLAS DE UNA O VARIAS TABLAS.- Se especifican valores posibles para registros completos según los valores acumulados registros previos o por valores existentes en registros de otras tablas.

La implementación de la cardinalidad del mapeo resultante en el modelo será una de las restricciones importantes que el sistema debe considerar. La programación de todas estas restricciones regularmente corre a cuenta de un programador especializado (que pudiera ser el DBA), mediante la adición de módulos al sistema.

Normalización de base de datos

Normalizar es un proceso que clasifica relaciones, objetos, formas de relación y demás elementos en grupos, en base a las características que cada uno posee. Si se identifican ciertas reglas, se aplica una categoría; si se definen otras reglas, se aplicará otra categoría. Busca eliminar la redundancia de las bases de datos, obteniendo las denominadas formas normales correspondientes. Este

proceso sigue normas bien definidas y reversibles, estas normas son aplicables durante el proceso de diseño de una base de datos.

Después de diseñar el diagrama Entidad-Relación el siguiente paso para el diseño de la base de datos es la Normalización que consiste en examinar los datos que se encuentran agrupados en una tabla hasta reemplazarlos por varias tablas que resultan ser simples y predecibles haciéndolas más fáciles de manejar. Este proceso se realiza por las siguientes razones:

- Estructurar los datos de tal manera que se pueda establecer fácilmente la relación entre los datos.
- Facilitar la recuperación de los datos para satisfacer las necesidades de información.
- Facilitar el mantenimiento de los datos (altas, bajas y cambios).
- Reducir la posibilidad de reorganizar o reestructurar la base de datos ante nuevas necesidades de almacenamiento de información.
- Reducir la posibilidad de redundancia e inconsistencias en los datos.

Para realizar este proceso se tienen cuatro pasos:

- Estado inicial
- Primera forma normal
- Segunda forma normal
- Tercera forma normal

Estado Inicial

En este paso se listan todos los datos del sistema, describiendo las tablas, llaves primarias y llaves externas que serán útiles para hacer los enlaces entre las tablas. Se examinan aquellos datos que tienen el resultado de un cálculo porque regularmente éstos no se almacenan en la base de datos, como sería el total de sueldos por departamento.

Primera forma normal

En la primera forma normal donde se eliminan las columnas que son redundantes en la misma fila y se crea una o más tablas y relaciones con los datos eliminados de la tabla original. Se alcanza la primera forma normal cuando ya no existen grupos de datos que se repiten.

Segunda forma normal

Se aplica para tablas cuya llave primaria es compuesta y se eliminan las columnas que depende sólo de una parte de la llave creando una o varias tablas con los datos que se removieron dejando únicamente los datos dependientes.

Tercera forma normal

Para que los datos se encuentren en tercera forma normal, las columnas no llave, no deben ser dependientes transitivos de la llave primaria. En una tabla se dice que tiene dependencia transitiva si:

La columna A es dependiente de la columna B y
la columna B depende de la columna C, entonces
la columna A depende de la columna C, siendo C la llave primaria.

Este tipo de dependencia puede provocar problemas si no se elimina, ya que de desconocer tal característica de los datos se pueden eliminar datos inapropiadamente, como sería el caso si se elimina C siendo A y B dependientes de C, también se borrarán.

Se han definido el ciclo de vida y la metodología estructurada ha seguir para el desarrollo del sistema de software del presente trabajo de tesis.

Cada uno de los siguientes capítulos que se contemplan en este trabajo corresponde con una de las fases del ciclo de vida. El siguiente capítulo se inicia con la primera etapa del ciclo de vida: "Planeación del sistema".

Planeación del sistema

Objetivo
Justificación
Propuesta de solución
Alcances y limitaciones
Estimación de costos
Costo de desarrollo
Costo de Operación
Beneficios del sistema propuesto

Como se ha visto en el capítulo anterior, el continuo problema con la piratería de software se traduce en importantes pérdidas de empleos, sueldos, ingresos por impuestos y una potencial barrera para la innovación y desarrollo de productos en todo el mundo. Es por ello, que nosotros como desarrolladores de software nos hemos visto en la necesidad de implementar un sistema que nos ayude a evitar que la piratería de los sistemas siga en aumento.

Para entender cual es la finalidad, funcionamiento y necesidades del sistema a desarrollar, se comienza con la planeación del sistema, determinando en primer lugar el objetivo a alcanzar al finalizar el presente proyecto de tesis.

Objetivo

Desarrollar un sistema integral para la generación y control de licencias de uso que permita la activación de software vía Internet, protegiendo así al software contra copias ilegales.

Justificación

El interés original que motivo a la realización de este sistema, es proporcionar una opción más a la industria del software para combatir la piratería, es decir, este sistema se presenta como una alternativa más que se añade a los métodos de protección de software, debido a que cada vez esta industria se ve más indefensa ante la piratería.

La piratería, es realmente una infracción sobre los derechos de propiedad intelectual de un usuario o compañía, no sólo afecta a la industria del software sino también a las industrias de música, cine, textiles, etc.

La reproducción no autorizada de software está proliferando cada día más, incluso se ha llegado a convertir en una amenaza empresarial a nivel internacional.

La reducción de ventas de software legítimo da como resultado la pérdida de impuestos y el descenso del nivel de empleo. La piratería de software obstaculiza enormemente el desarrollo de comunidades locales de software. Si el fabricante no puede vender sus productos en el mercado legal, no tendrá ningún incentivo para continuar desarrollando.

Muchos fabricantes de software incluso se niegan a entrar en mercados en los que el índice de piratería es muy alto, porque saben que no podrán recuperar sus inversiones para el desarrollo de nuevos productos.

La industria del software está comprometida a ayudar a los gobiernos en el reto de mejorar y hacer cumplir las leyes de propiedad intelectual y educar al público acerca de la importancia de los derechos de propiedad intelectual en el software. Ante esta realidad, el sistema a diseñar constituye una respuesta a la necesidad de desarrollar nuevos métodos que nos permitan evitar que la piratería de software aumente.

Cabe señalar que a nivel nacional los métodos existentes para combatir la piratería no han sido suficientes, este sistema no pretende ser una solución al problema pero sí una alternativa más para combatirlo.

Por otro lado dicho sistema trae grandes beneficios económicos a la industria del software evitando las pérdidas millonarias que día con día se generan. Otro beneficio es que el sistema está basado únicamente en software y la empresa que lo utilice en sus productos de software, no tiene que invertir en ningún dispositivo de hardware (por ejemplo un dispositivo de protección en un puerto del equipo de cómputo) adicional para su funcionamiento, el cual incrementaría el precio final del producto.

Propuesta de Solución.

Como se ha mencionado con anterioridad se propone diseñar e implantar un sistema que permita evitar la reproducción de copias ilegales de un software de aplicación.

Dicho sistema guarda la información en bases de datos relacionales, ya que de esta forma la información puede ser almacenada en más de una tabla, por lo que, de esta manera dicha información es recuperada de forma fácil y eficiente.

El sistema propuesto opera de la siguiente manera:

La empresa de desarrollo de software, que implante este sistema de protección de software a sus productos, dispone de un programa que se encarga de generar el número de serie único para cada producto, así mismo genera la impresión del mismo en una etiqueta que será anexada al producto. Este programa a su vez almacena cada número de serie generado en dos bases de datos, una ubicada en un equipo local y otra alojada en un sitio de Internet, asignándole la licencia de uso correspondiente. Una vez que el usuario final del producto ha completado la instalación, al ejecutar la aplicación por primera ocasión se iniciará el proceso de activación a través de un asistente que lo guiará durante todo el proceso, dicho asistente procede de la siguiente manera:

1. Se le solicita se ingrese el número de serie que acompaña al paquete de software, escrito en una etiqueta.

2. Se le pregunta al usuario si dispone de una conexión a Internet, de ser así el asistente verifica el número de serie en la base de datos alojada en el sitio de Internet.
3. Se muestra en pantalla un formulario de registro que el usuario debe llenar con sus datos personales.
4. Después de completar el registro se muestra en pantalla el código de usuario asignado, con el cual puede activar actualizaciones, activar licencias adicionales adquiridas y consultar desde Internet el estado de las mismas.
5. Finalmente el asistente de activación libera el software para su uso exclusivo en el equipo instalado.

En el momento de la liberación del software vía Internet se almacena en la base de datos ubicada en el sitio de Internet, la fecha y hora en que fue concedida la licencia de uso y así es marcada como otorgada. El usuario puede instalar el software en otro equipo pero al momento de activación se le niega la licencia de uso ya que el asistente de activación comprueba que ya fue otorgada. Por lo tanto el usuario debe adquirir una licencia de uso adicional para cada equipo donde desee utilizar el software.

Si el usuario no dispone de una conexión a Internet en el equipo de instalación, el asistente de activación le genera y muestra en pantalla un código de equipo único, así mismo, le solicita que ingrese la clave de acceso, la cual puede obtener vía página Web (desde cualquier otro equipo con acceso a Internet) o vía telefónica, en ambos casos debe el usuario proporcionar el número de serie y el código de equipo que le muestra en pantalla el asistente.

Al igual que la activación del software vía Internet, en el caso de la obtención de la clave de acceso vía página Web, se almacena en la base de datos ubicada en el sitio de Internet, la fecha y hora en que fue concedida la licencia de uso y por lo tanto es marcada como otorgada.

Para la obtención de la clave de acceso vía telefónica, el usuario debe comunicarse con el distribuidor del producto adquirido y este debe de contar con acceso a Internet para poder obtener la clave de acceso vía página Web y finalmente proporcionarla al usuario.

Al adquirir el usuario final una licencia adicional del producto, la empresa que desarrolla el mismo, se encarga de agregar en la base de datos alojada en el sitio de Internet, un registro indicando la disponibilidad de una licencia de uso para el número de serie correspondiente al producto del usuario. De esta forma obligamos al usuario a poseer una licencia de uso para cada equipo donde desee emplear el software y con ello la empresa que lo desarrolla lleva un registro de las licencias de uso concedidas a sus usuarios.

La empresa de desarrollo debe establecer en su contrato de licencia de uso de software, la cantidad de veces que un usuario puede solicitar la reposición de licencia de uso, en el caso de que formatee su disco duro, cambie de disco duro o cambie de equipo.

Alcances y limitaciones.

Los alcances de este sistema dependen en gran medida de la infraestructura tecnológica utilizada en su diseño y desarrollo, por otro lado la empresa que desee implantar este sistema de protección de software, no tiene que disponer de una gran tecnología computacional por lo que no solo es un planteamiento teórico sino práctico lo que logra que su implantación pueda llevarse a cabo.

- El sistema abarca desde la generación del número de serie único para cada producto vendido, alta en Internet, seguimiento del mismo de acuerdo a número de licencias disponibles, reposiciones y usuario.
- Control sobre productos vendidos, devoluciones, número de licencias por producto, versiones y distribuidores.
- Seguridad en el acceso al sistema mediante la autenticación de usuario.
- Implementación de algoritmos complejos en la generación de números de serie.
- El sistema permite la emisión de reportes sobre:
 - Productos vendidos
 - Productos ya registrados
 - Productos por distribuidor
 - Devoluciones
- El sistema tiene un ambiente de trabajo para el usuario final de fácil manejo, logrando así, que el sistema sea lo más amigable posible y pueda aprovecharse al máximo su capacidad.

- Notificación de registro de software y recuperación de datos de usuario final.
- El usuario final del producto puede activar su software de forma fácil, adquirir licencias adicionales y consultar vía Internet el estado de su licencia de uso.

Las limitaciones del sistema se reducen a la disponibilidad de la empresa para contar con los mínimos requerimientos de operación del sistema, los cuales se mencionan en la fase de implantación.

Por otro lado el sistema no dispone de:

- Una tienda virtual donde el usuario adquiera licencias adicionales y estas sean agregadas a su registro de licencias disponibles para su uso inmediato.
- Módulo de reposición inmediata de licencias de uso por lo que el usuario debe solicitar vía telefónica o E-mail la reposición de la misma.

Estimación de Costos.

Esta es sin duda una tarea difícil al momento de realizar sistemas. Se debe de tener la suficiente visión para saber cuales son los costos de desarrollo y poder visualizar con anticipación cuáles son los eventos que se presentan a lo largo del desarrollo. Es recomendable realizar un estudio preliminar y realizar una revisión de requisitos del sistema, como son: capacidad del programador, complejidad del producto, tamaño del programa, tiempo disponible, confiabilidad requerida, nivel tecnológico, etc.

Costos de desarrollo

El costo de las herramientas de software para la realización del sistema planeado, es nulo, debido a que ya se cuenta con las herramientas de desarrollo necesarias. Algunas de estas herramientas habían sido adquiridas con anterioridad por los autores de este proyecto de tesis y otras son gratuitas, en el capítulo 4 se definen cuales son estas herramientas.

En lo referente al costo de hardware igualmente es nulo, ya que se dispone de un equipo de cómputo que cumple con los requisitos mínimos para soportar las herramientas de diseño, programación y evaluación del sistema a desarrollar. El equipo de computo a utilizar posee un procesador Pentium II a 450 MHz, 128 Mb en RAM y 10 GB de disco duro.

No se tiene un costo por capacitación para el uso de las herramientas de desarrollo, ya que se posee un amplio conocimiento de las mismas, conocimiento que fue adquirido con anterioridad mediante cursos y en el ambiente laboral.

El costo de programación es de 300 pesos por hora. Estimando que el tiempo de desarrollo fuera de 60 días con jornada de 8 horas, el costo aproximado de programación sería de \$144,000.00 pesos.

Costos de operación

Los costos de operación para la empresa que implante el sistema propuesto, son prácticamente nulos debido a que solo tiene que proporcionar a su departamento de desarrollo el componente de software que constituye el Subsistema Asistente de Activación de Software, dicho componente debe ser agregado a su aplicación y recompilar.

A continuación se listan los costos de operación:

- Renta de sitio en Internet con disponibilidad de alojar páginas ASP.
- Equipo de cómputo Pentium II a 450 MHz o superior.
- Etiquetas de impresión para el Número de Serie de cada producto a distribuir.
- Conexión de acceso a Internet.
- Personal capacitado para la operación de equipos de cómputo.

Como puede observarse los costos de operación son mínimos, además de que la mayoría de las empresas dedicadas al desarrollo de software, cuentan ya con algunos de ellos como son: equipo de cómputo, personal, acceso a Internet y probablemente un sitio en Internet; por lo que sólo tiene que adquirir las etiquetas para la impresión de número de serie.

Es importante mencionar que el costo en relación tiempo de aprendizaje de operación del sistema es muy breve debido a la sencillez de todos los subsistemas que lo conforman, es decir, no existe complejidad alguna en la operación del mismo con lo cual el tiempo de aprendizaje es mínimo.

Beneficios del sistema.

- El gasto correspondiente a la adquisición de este sistema integral, por parte de la empresa desarrolladora de software, es sin lugar a duda mucho menor que las pérdidas por copias ilegales de sus aplicaciones.
- En corto plazo puede recuperar lo invertido y así disponer de más recursos para investigación, desarrollo, mejores productos y más actualizaciones.
- Así mismo proporcionar al usuario una adecuada garantía, asistencia técnica, asesoría post-venta y acceso a versiones mejoradas a bajos precios.
- Los beneficios generados al disponer de un sistema de protección de software serán no tan sólo para la empresa de desarrollo sino también para el usuario final, ya que este se beneficia del uso de software legal que impacta positivamente tanto en la economía local como en la nacional. El usuario tiene la seguridad de que el pago realizado por software adquirido esta dentro de un mercado de legalidad y con ello se brinda empleo a miles de personas.
- La interfase gráfica para el usuario final asegura un uso intuitivo, guiándolo rápida y fácilmente a través del proceso de registro y activación de software.
- La empresa de desarrollo puede llevar un control de sus productos y licencias de uso dentro y fuera de la misma.

Una vez que se ha completado la planeación del sistema se procede, en el siguiente capítulo, a realizar una especificación de los requerimientos del sistema, se definen los subsistemas que lo integran, diagramas de flujo, diagramas entidad relación y diccionarios de datos de las bases de datos utilizadas.

Análisis del sistema

Determinación de requerimientos
Subsistemas que integran el sistema
Diagramas de flujo
Usuarios del sistema
Diagramas de flujo estructurado
Estructura de Base de datos Local
 Diagrama Entidad Relación
Estructura de Base de datos en Internet
 Diagrama Entidad Relación

En el capítulo anterior se ha realizado la planeación del sistema que propone el presente trabajo de tesis, definiendo su objetivo, justificación, propuesta de solución, alcances, limitaciones y costos. En este capítulo se hace un análisis del sistema y con ello se determinan los requerimientos que el sistema deberá cumplir.

Análisis del sistema.

Como desarrolladores de software y en base a la información anterior decidimos diseñar e implantar un sistema para proteger de la piratería los programas de aplicación, básicamente se requiere que la aplicación protegida, después de ser instalada por el usuario final, sólo una vez pueda ser activada; de tal manera que aunque la aplicación se instale en más de un equipo, no podrá ser activada para su uso. Además es indispensable que la aplicación ya activada no pueda ser redistribuida por el usuario a terceras personas.

Determinamos que la mejor forma de identificar cada paquete de software distribuido es, asignándole un número de serie único, como lo hace la gran mayoría de las empresas desarrolladoras de software. Pero dicho número de serie servirá únicamente para verificar la autenticidad del paquete, no para activar directamente el programa de aplicación. Esta determinación se realizó con base a la facilidad con que han sido pirateados los programas de aplicación, que con solo disponer de un número de serie, el usuario final puede activar la aplicación en cuantas computadoras quiera, haciendo caso omiso del contrato de licencia que le estipula que debe pagar una licencia de uso por cada equipo adicional donde utilice la aplicación.

En otras palabras, el usuario final del programa de aplicación debe obtener la licencia de uso para la activación del mismo. Por lo que tiene que solicitarla directamente a la empresa que desarrollo el software y verificar en una base de datos, si aún dispone de licencias de uso para el número de serie proporcionado por el usuario. Entonces, se necesita de un medio de comunicación entre el programa de aplicación y la empresa que lo desarrollo, para ello se determino que Internet sería el mejor medio de comunicación debido a su bajo costo y disponibilidad todos los días del año las 24 horas del día.

Al determinar la utilización de Internet como medio de comunicación para la obtención de la licencia de uso de software de aplicación y consecuentemente permita la activación del mismo, el usuario final podrá disponer de su licencia cuando lo desee y solamente en una ocasión.

En este punto es importante mencionar que se alojará una base de datos en un sitio de Internet, que contendrá la información respectiva a los números de serie distribuidos y la cantidad de licencias de uso disponibles para cada uno de ellos.

Por lo tanto, el sistema a implantar debe encargarse de administrar las licencias de uso del software para cada paquete de software distribuido y debe de cumplir con las siguientes características que son esenciales para cualquier empresa de desarrollo de software:

- Módulo de productos donde la empresa pueda incluir los nombres de sus aplicaciones y con ello pueda organizar sus paquetes de venta.
- Módulo para crear y definir paquetes de software, es decir, los productos que conforman un paquete, sus respectivas licencias de uso y descripción del mismo.
- Directorio de los distribuidores de la empresa con datos tales como: Nombre, Dirección, RFC, Nombre de Contacto, Puesto, Teléfono, etc.
- Se requiere de un módulo donde capture el número de inventario del producto ya empaquetado que saldrá de la empresa, distribuidor al que es enviado, genere el número de serie único que se envía al Servidor de licencias de uso en Internet, además de imprimirse la etiqueta correspondiente que será anexada dentro del producto antes de empaquetarse.
- Módulo que permita consultar los productos que han salido de la empresa, mostrándose datos como: paquete, contenido, distribuidor y si es o no de devolución. En el caso de que el producto sea devuelto, llevar un registro de devoluciones. Toda devolución podrá ser redistribuida respetando el número de inventario correspondiente a dicho producto.
- Módulo para asignación de licencias adicionales o de reposición que el usuario final solicite. Así mismo envió de las mismas al Servidor de licencias de uso en Internet para su disponibilidad.
- Generador de informes con relación a productos, licencias, paquetes y distribuidores.

Por otra parte las características del sistema para los usuarios finales del programa de aplicación de la empresa desarrolladora del software, son:

Contar con un asistente de activación de software, que de forma amigable y sencilla guíe paso a paso al usuario final durante el proceso de activación, el asistente debe ejecutarse al iniciar la aplicación mientras esta no haya sido activada.

El asistente de activación debe:

1. Solicitar se ingrese el número de serie del producto y validar su autenticidad.
2. Verificar que exista conexión a Internet en el equipo de instalación.
3. Proporcionar al usuario un código de equipo único en el caso de no haber detectado una conexión a Internet activa.
4. Hacer las validaciones pertinentes que permitan detectar alguna anomalía con los datos del software proporcionados.
5. Solicitar al usuario mediante un formulario, sus datos personales para el registro del software y enviarlos a la base de datos alojada en el sitio de Internet.
6. Devolver un código de usuario para su identificación y posteriores actualizaciones, reposiciones y consultas.
7. Comprobar en la base de datos alojada en Internet, la disponibilidad de licencia de uso y asignarla al usuario en caso de contar con alguna.
8. Finalmente debe activar el software para su uso.

Concluido el proceso satisfactoriamente, el asistente de activación debe liberar el software para su uso y no volverse a mostrar al iniciar la aplicación. La aplicación debe cerrarse si no se obtuvo una licencia de uso.

Hasta este momento se han determinado cuales son las características que debe cumplir el sistema, ahora se determinan los requerimientos y se definen los subsistemas que integran al sistema de acuerdo al análisis realizado.

Determinación de requerimientos

Debido a que el sistema debe ser operado tanto por la empresa que desarrolla la aplicación a proteger contra la piratería, como por los usuarios de la misma, el sistema queda dividido de la siguiente manera:

- a) Subsistemas que integran el sistema.
- b) Información que se maneja en cada subsistema.
- c) Diagramas de flujo
- d) Usuarios.

Permitiéndose así un manejo eficiente y seguro del mismo, así como también, de esta forma se pretende cumplir con todas y cada una de las características analizadas para lograr su desarrollo.

Subsistemas que integran el sistema:

- Generador y administrador de licencias de uso de software.
- Asistente de activación de software.
- Servidor de licencias de uso de software en Internet.
- Registro, obtención y consulta de licencia de uso en Internet.

Subsistema Generador y administrador de licencias de uso de software.

Este subsistema es para uso exclusivo de la empresa que implantará este sistema de protección en sus aplicaciones. Emplea una base de datos relacional sobre la cual se lleva el registro de:

- Productos de software que desarrolle la empresa.
- Paquetes de software conformados por uno o más productos.
- Distribuidores de software.
- Salidas de Paquetes de software.
- Licencias de uso de software.
- Licencias de uso de software adicionales.
- Usuario final de paquete.
- Devoluciones de paquetes.

El objetivo de este subsistema es el de llevar el registro de paquetes de software a distribuir, a través de un entorno visual se puede almacenar en la base de datos local la información correspondiente a cada paquete:

- Número de Inventario
- Contenido del paquete
- Número de serie asignado
- Licencias de uso correspondientes
- Licencias de uso adicionales
- Licencias de uso por reposición
- Distribuidor al que fue enviado
- Descripción del paquete

El subsistema genera el número de serie que identifica de manera única a cada producto, envía de forma automática el mismo a la base de datos alojada en el sitio de Internet de la empresa desarrolladora de software y asigna la(s) licencia(s) de uso disponibles para cada producto distribuido. El usuario de este subsistema tiene la facultad de agregar, editar y eliminar licencias de uso de software, manteniendo en sincronía la información con la base de datos alojada en el sitio de Internet, para que los usuarios finales de la aplicación puedan registrar su software adquirido y obtener la licencia de uso correspondiente para la activación del mismo.

El subsistema recupera de la base de datos de Internet los datos de usuario de los paquetes ya registrados, así mismo las licencias de uso concedidas, junto con ello la fecha y hora en que acontecieron, de esta manera se mantiene un registro de información local para su consulta y análisis sin necesidad de una conexión permanente a Internet incrementando la velocidad en la obtención de la información. Se lleva un control de distribución de paquetes mediante la utilización de un directorio de distribuidores de software, por cada distribuidor se tiene un registro de información tal como: Nombre del distribuidor, dirección, contacto, entre otros. La empresa puede realizar consultas sobre distribuidores, pudiendo obtener información sobre ventas por distribuidor, software ya registrado por cada distribuidor, entre otras consultas.

El sistema establece la comunicación con la base de datos en Internet a través de páginas Web las cuales se encargan de realizar la conexión utilizando el controlador de base de datos apropiado con el sistema de base de datos utilizado, en la etapa de desarrollo se define el manejador de base de datos a emplear.

Subsistema Asistente de activación de software.

Conformado por un componente de software con propiedades y métodos, encapsulado en una librería como un componente reutilizable; podrá ser incorporado al programa de aplicación que la empresa desarrolle. Los desarrolladores de la empresa deben proporcionar los valores de las propiedades de la librería, las cuales constituyen los elementos para la activación del software. Así mismo proporcionarle la dirección HTTP del sitio en Internet donde la empresa tiene la base de datos que forma parte del subsistema servidor de licencias de uso. La librería establece comunicación con el servidor de licencias en Internet para la validación del número de serie, licencias de uso disponibles, registro de usuario final y recuperar el código de usuario de la copia de software.

La comunicación del asistente de activación con el servidor de licencias en Internet es mediante la consulta de páginas Web, las cuales se encargan de realizar las operaciones de inserción, edición, borrado y consulta de los registros contenidos en la base de datos del servidor de licencias.

El asistente envía las peticiones hacia paginas Web específicas para cada operación, básicamente a través de la dirección URL son enviados los criterios necesarios para la elaboración de sentencias de consulta dentro de la página, la cual se encarga de redireccionar hacia una nueva página y de ella obtener los resultados de la operación solicitada o bien la confirmación de inserción, actualización o eliminación.

A través de un entorno visual guía al usuario final durante el proceso de registro y activación del software adquirido. Haciendo uso de botones de avance y

retroceso, el usuario puede navegar hacia delante y hacia atrás en cada paso del proceso de activación. Cada paso dispone de una leyenda describiendo las acciones que debe realizar el usuario y solicitudes de información mediante cuadros de texto.

Servidor de licencias de uso de software en Internet.

Constituido esencialmente por un sitio en Internet. En este sitio se aloja una base de datos relacional sobre la cual se lleva el registro de número de serie de paquetes de software ya distribuidos, licencias de uso correspondientes a cada paquete, licencias adicionales, licencias por reposición y los datos de registro del usuario final del software. Este servidor puede ser accesado a través de páginas Web que brindan el servicio de conexión a la base de datos, inserción, modificación, borrado y consulta de información. El acceso es controlado mediante una autenticación conformada por el número de serie del producto y el código de usuario que registro el producto.

El servidor cuenta con algoritmos necesarios para la decodificación de los números de serie y licencias de uso de software. Mantiene en sincronía con la base de datos ubicada en la empresa la información referente a paquetes de software ya registrados, los datos del usuario final e información de licencias concedidas a cada paquete. La seguridad en el sitio es primordial para evitar el robo de información.

Subsistema de Registro, obtención y consulta de licencia de uso en Internet.

Constituido por un sitio en Internet que realiza una comunicación directa con el servidor de licencias en Internet, mediante paginas Web permite al usuario final, el registro de su paquete de software y la obtención de las licencias de uso correspondientes. Este subsistema constituye la alternativa de activación de software en el caso de que el usuario final no disponga de una conexión a Internet en el equipo final de instalación. El usuario puede acceder directamente al sitio desde cualquier equipo con acceso a Internet, o bien puede contactar vía telefónica a su distribuidor. El acceso al sitio esta restringido, sólo usuarios que dispongan de un número de serie válido podrán acceder al mismo. Si el usuario no ha registrado aún su paquete de software, después de ingresar su número de serie se le redirecciona hacia un formulario de registro, de lo contrario debe proporcionar su código de usuario que le fue proporcionado después de su registro. Después del proceso de registro el usuario final puede consultar licencias de uso disponibles y concedidas. Tendrá acceso a la actualización de sus datos de registro los cuales constituyen información valiosa para la empresa como es el caso del correo electrónico del usuario, a través del cual se le podrá enviar información sobre nuevos productos, actualizaciones, soporte técnico, entre otros.

Una vez analizado cada uno de los subsistemas que integran al sistema final, a continuación se muestran los diagramas de flujo para cada uno de ellos.

Diagramas de Flujo

A medida que la información se mueve a través del sistema, está es modificada por una serie de transformaciones. Los diagramas de flujo de datos (DFD) siguientes representan el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida.

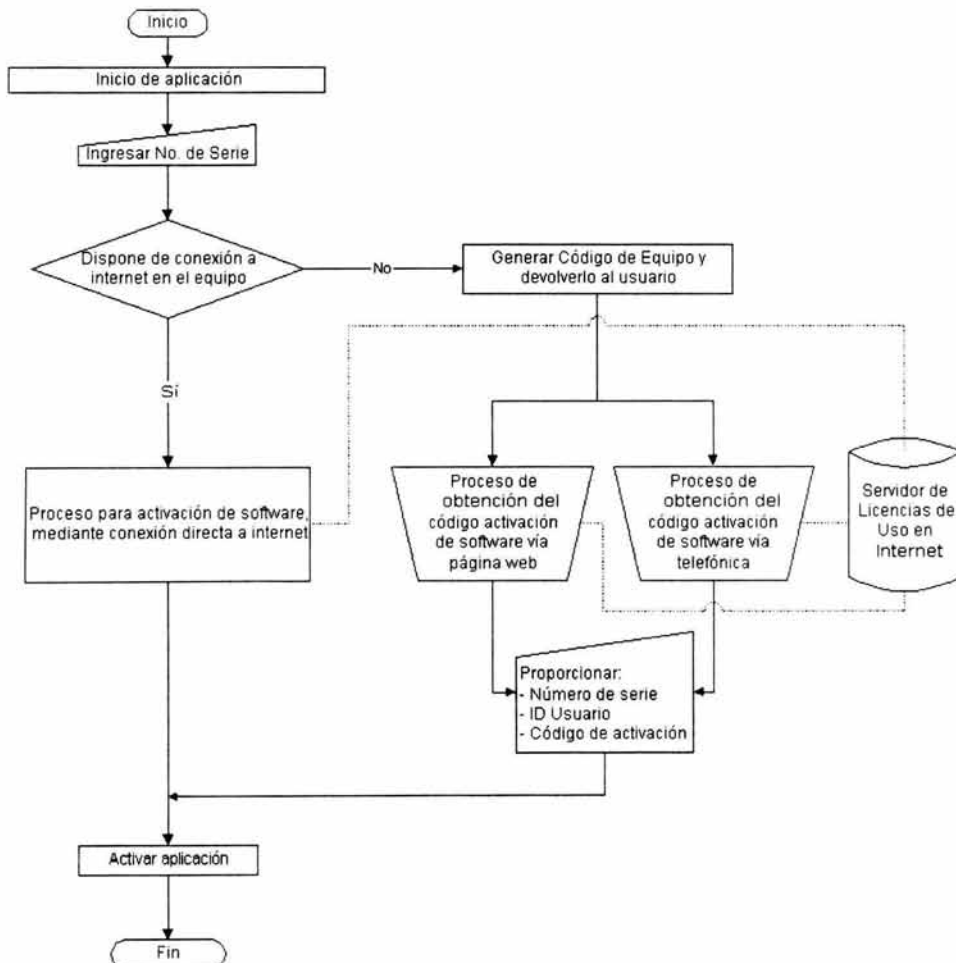


Figura III.1 formas de registro

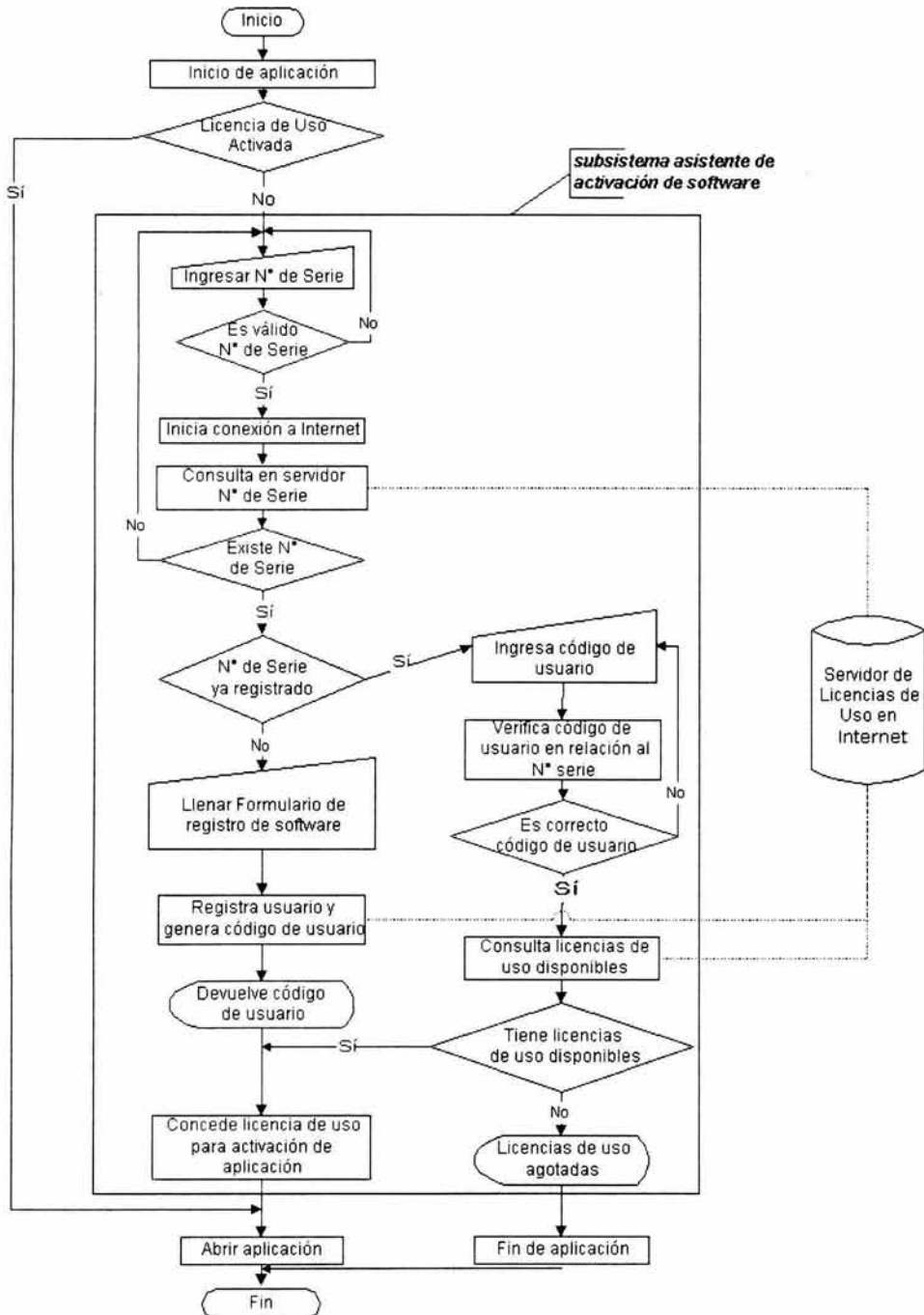


Figura III.2 Proceso de registro y activación de software mediante conexión directa a Internet

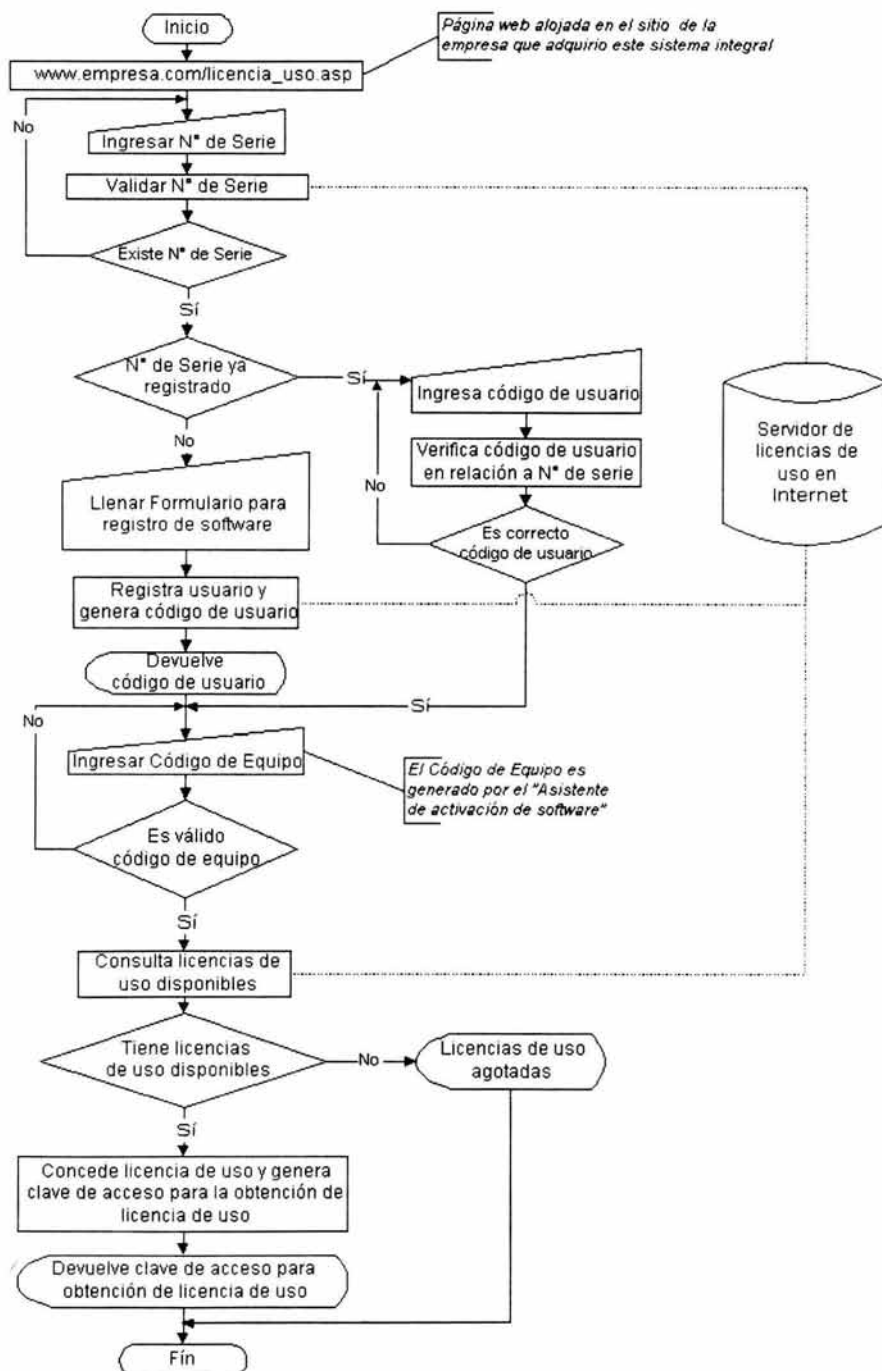


Figura III.3 Proceso de registro y activación de software vía página web

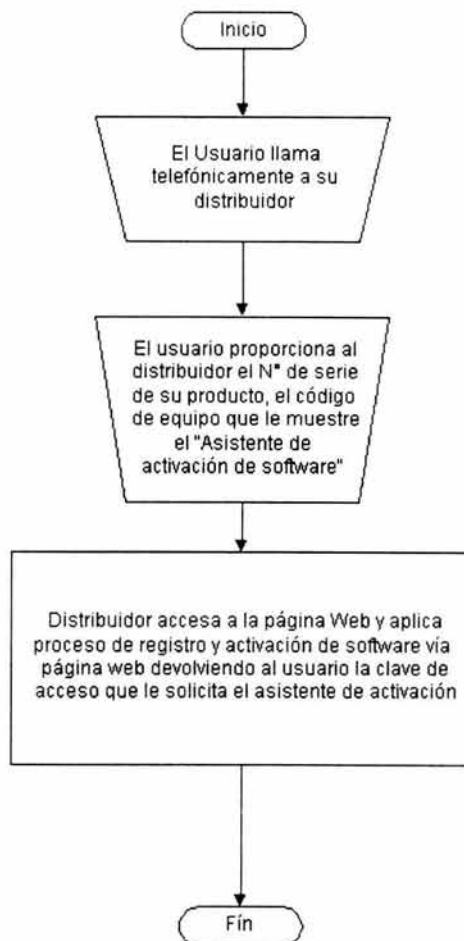


Figura III.4 Proceso de registro y activación vía telefónica

Usuarios del sistema.

Se definieron los usuarios de acuerdo a cada subsistema que integran al sistema global.

Subsistema Generador y administrador de licencias de uso de software:

-Personal designado por la empresa para su operación: Este personal debe poseer los conocimientos mínimos de operación de un sistema de cómputo. Además de ser de absoluta confianza para la empresa ya que es la persona encargada de generar el número de serie de sus productos.

Subsistema Asistente de activación de software:

-Programadores de la empresa: Generan la versión liberada de las aplicaciones que desarrolle la empresa. Encargados de insertar y hacer uso de las propiedades y métodos del componente del software que constituye el Asistente de activación de software.

-Usuario Final de la aplicación que distribuya la empresa desarrolladora: El usuario final puede activar y registrar su software a través del asistente que lo guía paso a paso en el proceso.

Servidor de licencias de uso de software en Internet.

-Los usuarios de este servidor son el resto de los subsistemas, es decir, el generador de licencias, el asistente de activación de software y el subsistema de registro, obtención y consulta de licencia de uso en Internet. Debido a que gran parte de la información procesada por dichos subsistemas se encuentra almacenada en el servidor de licencias.

Subsistema de Registro, obtención y consulta de licencia de uso en Internet.

-Personal de asistencia telefónica de la empresa y distribuidores.

-Usuarios Finales

Una vez definido el flujo de la información y usuarios para el sistema, se continua con el diseño de los diagramas estructurados.

Diagramas Estructurados

En el primer capítulo se determinó que el desarrollo del sistema propuesto está definido a través de una Metodología estructurada, por lo que a continuación se muestran los siguientes diagramas estructurados.

La función global del sistema se representa como una transformación sencilla de la información, representada en la figura como una burbuja. Una o más entradas representadas como flechas con etiqueta, conducen la transformación para producir la información de salida. El sistema acepta entradas de distintas formas; aplica un hardware, software y elementos humanos para transformar la entrada en salida; y produce una salida en distintas formas. La transformación puede comprender desde una sencilla comparación lógica hasta un complejo algoritmo numérico.

NIVEL 0 Modelo de contexto

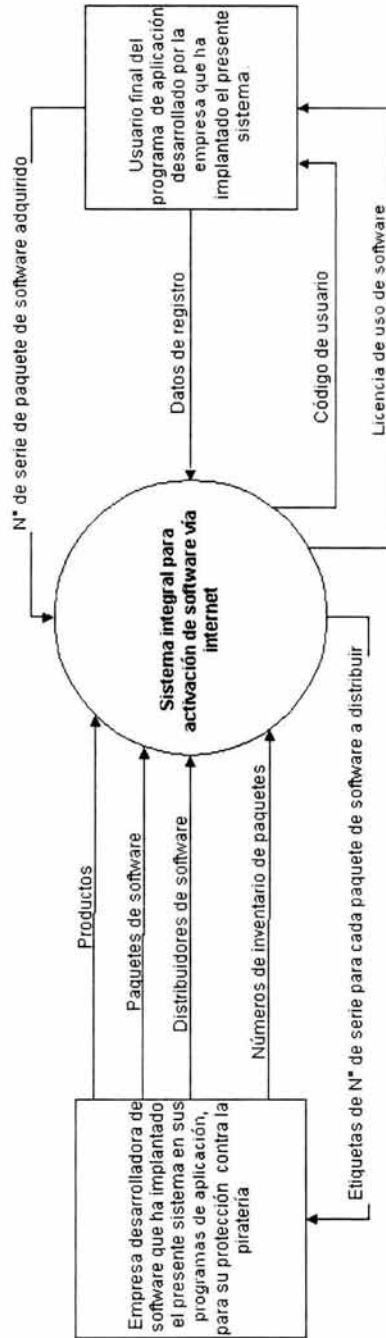


Figura III.5 Diagrama estructurado de nivel 0

NIVEL 1

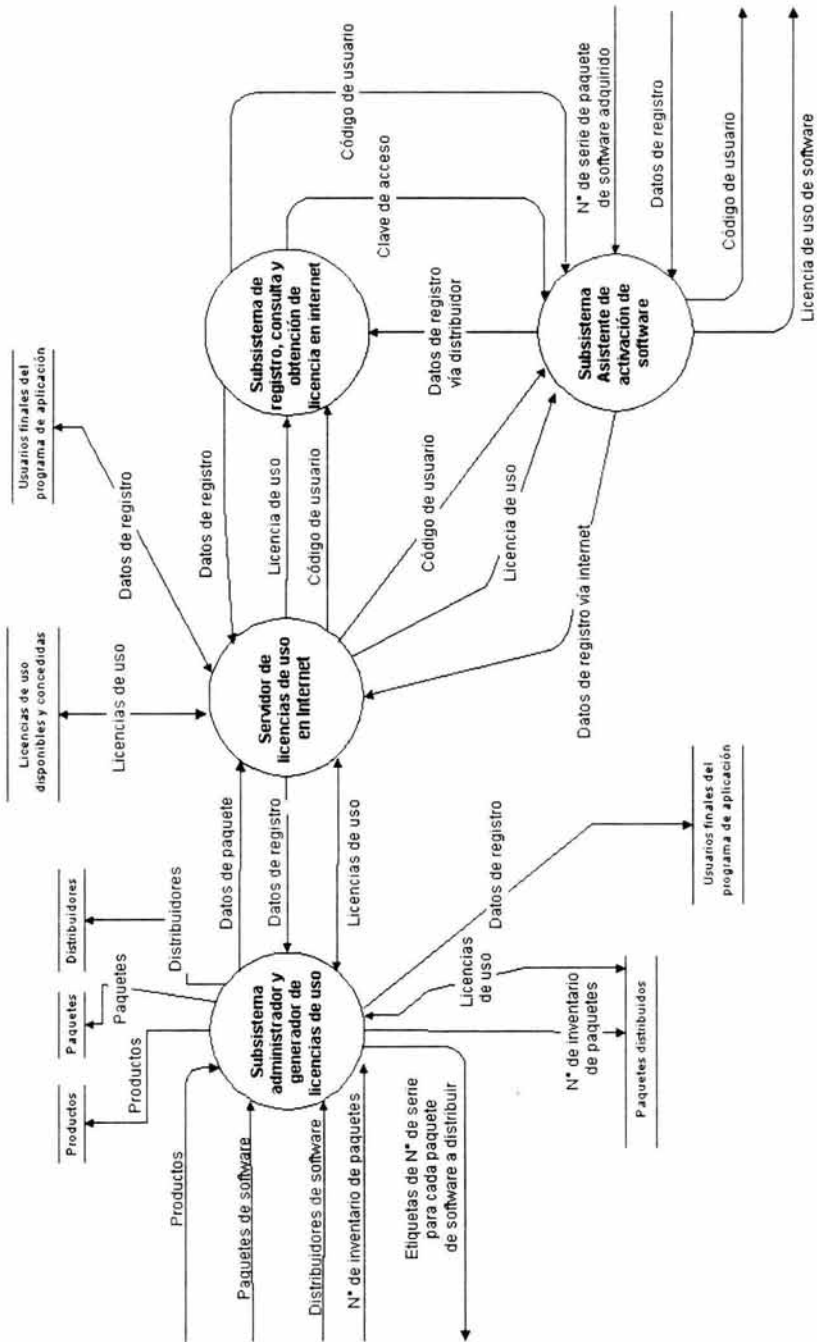


Figura III.6 Diagrama estructurado de nivel 1

La figura III.5 presenta el diagrama estructurado de nivel cero ó también llamado “Modelo de contexto” en el se representa el funcionamiento general del sistema como una única transformación de información. Las entidades externas originan una o más entradas de información, que aparecen como flechas etiquetadas, y la información de salida se dirige hacia las entidades externas.

De acuerdo al análisis y determinación de los requerimientos estudiados anteriormente, se define a continuación la estructura de las dos bases de datos a utilizar, es decir, la base de datos ubicada localmente y una segunda alojada en el sitio de Internet.

Estructura de la Base de Datos Local.

Esta base de datos forma parte del subsistema administrador y generador de licencias de uso, dicha base de datos lleva como nombre **BD_GL.mdb**. En ella se almacena información que es utilizada para cada uno de los subsistemas a desarrollar.

Tablas

CDs

Nombre del Campo	Tipo	Tamaño
CodCD	Long Integer	4
NumInventario	Long Integer	4
NumSerie	Text	20
FechaSalida	Date/Time	8
FechaPublicado	Text	50
ID_CDPaquete	Long Integer	4
ID_CDDistribuidor	Long Integer	4
ID_Usuario	Long Integer	4
Nota	Memo	-
Publicado	Yes/No	1
EsDevolucion	Yes/No	1
PorEliminar	Yes/No	1

LicsCD

ID_Lic	Long Integer	4
CodCD	Long Integer	4
KeyTipoLic	Long Integer	4
FechaAsignada	Date/Time	8
Concedida	Yes/No	1
FechaConcedida	Date/Time	8
DePaquete	Yes/No	1
EsAdicional	Yes/No	1
EsReposicion	Yes/No	1
Publicado	Yes/No	1
PorEliminar	Yes/No	1

KeyTipoLic		
KeyTipoLic	Long Integer	4
Producto	Text	50
Publicado	Yes/No	1
PorEliminar	Yes/No	1
CDPaquete		
ID_CDPaquete	Long Integer	4
CDPaquete	Text	100
NumVersion	Single	4
Descripcion	Memo	-
Publicado	Yes/No	1
PorEliminar	Yes/No	1
CDDetallePaquete		
ID_CDDetallePaquete	Long Integer	4
ID_CDPaquete	Long Integer	4
KeyTipoLic	Long Integer	4
CDDistribuidor		
ID_CDDistribuidor	Long Integer	4
CDDistribuidor	Text	100
CalleNumero	Text	50
Colonia	Text	50
CP	Text	6
Ciudad	Text	50
Estado	Text	50
Pais	Text	50
RFC	Text	50
Contacto	Text	50
Puesto	Text	50
Telefono	Text	50
Fax	Text	50
Email	Text	100
Otros	Memo	-
Usuarios		
ID_Usuario	Long Integer	4
Fecha	Date/Time	8
Nombre	Text	50
ApePaterno	Text	30
ApeMaterno	Text	30
CalleNumero	Text	70
Colonia	Text	70
CP	Text	10
Cd	Text	50
Edo	Text	50
Pais	Text	50
Lada	Text	15
Telefono	Text	25
Extension	Text	15
Email	Text	100
Notas	Memo	-

CDsDevolucion

ID_CDsDevolucion	Long Integer	4
CodCD	Long Integer	4
NumSerie	Text	20
NumInventario	Long Integer	4
FechaSalida	Date/Time	8
FechaDevolucion	Date/Time	8
ID_CDPaquete	Long Integer	4
ID_CDDistribuidor	Long Integer	4
ID_Usuario	Long Integer	4
Licencias	Memo	-

Consultas Almacenadas**C_CDs**

Nombre del Campo	Tipo	Tamaño
CodCD	Long Integer	4
NumInventario	Long Integer	4
FechaSalida	Date/Time	8
ID_CDPaquete	Long Integer	4
ID_CDDistribuidor	Long Integer	4
Nota	Memo	N/A
EsDevolucion	Yes/No	1
NumVersion	Single	4
ID_CDEspecialidad	Long Integer	4

C_LicsCD

ID_Lic	Long Integer	4
CodCD	Long Integer	4
KeyTipoLic	Long Integer	4
Producto	Text	50
FechaAsignada	Date/Time	8
Concedida	Yes/No	1
FechaConcedida	Date/Time	8
EsReposicion	Yes/No	1
Publicado	Yes/No	1
PorEliminar	Yes/No	1
EsDevolucion	Yes/No	1

C_CDPaquete

ID_CDPaquete	Long Integer	4
NumVersion	Single	4

C_CDDetallePaquete

ID_CDPaquete	Long Integer	4
ID_CDDetallePaquete	Long Integer	4
KeyTipoLic	Long Integer	4
Producto	Text	50

C_CDUsr

ID_CD	Binary	70
CodCD	Long Integer	4
ID_CDDistribuidor	Long Integer	4
ID_Usuario	Long Integer	4
Fecha	Date/Time	8
Nombre	Text	50
ApePaterno	Text	30
ApeMaterno	Text	30
CalleNumero	Text	70
Colonia	Text	70
CP	Text	10
Cd	Text	50
Edo	Text	50
Pais	Text	50
Lada	Text	15
Telefono	Text	25
Extension	Text	15
Email	Text	100

C_CodCDUsuario

CodCD	Long Integer	4
ID_Usuario	Long Integer	4
Nombre	Text	50
ApePaterno	Text	30
ApeMaterno	Text	30
CalleNumero	Text	70
Colonia	Text	70
CP	Text	10
Cd	Text	50
Edo	Text	50
Pais	Text	50
Lada	Text	15
Telefono	Text	25
Extension	Text	15
Email	Text	100

C_CDDevolucion

NumInventario	Long Integer	4
FechaSalida	Date/Time	8
FechaDevolucion	Date/Time	8
ID_CDPaquete	Long Integer	4
CDPaquete	Text	100
NumVersion	Single	4
ID_CDDistribuidor	Long Integer	4
CDDistribuidor	Text	100

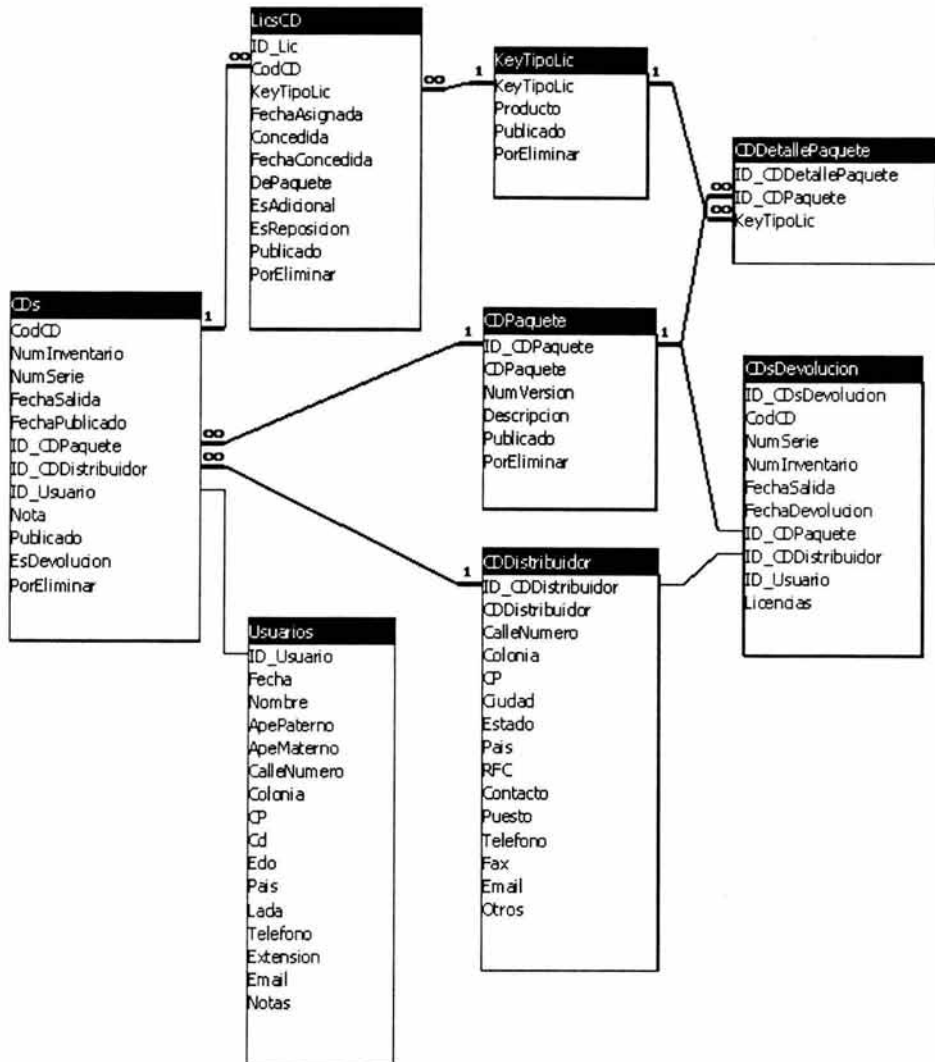


Figura III.7 Diagrama entidad-relación de la base de datos local

Estructura de la Base de Datos de Internet.

Esta base de datos forma parte del servidor de licencias de uso en Internet por lo tanto se encuentra alojada en el sitio de Internet de la empresa desarrolladora que implante este sistema de protección a sus programas de aplicación, dicha base de datos lleva como nombre **BD_LICS.mdb**. En ella se almacena información que es de utilidad para el Asistente de registro y para los usuarios finales de la aplicación.

Tablas

CDs

Nombre del Campo	Tipo	Tamaño
ID_CD	Long Integer	4
Fecha	Date/Time	8
CodCD	Long Integer	4
NoSerie	Text	20
ID_Paquete	Long Integer	4
ID_Usuario	Long Integer	4
NoVersion	Single	4

KeyTipoLic

KeyTipoLic	Long Integer	4
Producto	Text	50

LicsCD

ID_Lic	Long Integer	4
CodCD	Long Integer	4
KeyTipoLic	Long Integer	4
FechaAsignada	Date/Time	8
Concedida	Yes/No	1
FechaConcedida	Date/Time	8
DePaquete	Yes/No	1
EsAdicional	Yes/No	1
EsReposicion	Yes/No	1
Descargado	Yes/No	1

Paquetes

ID_Paquete	Long Integer	4
Paquete	Text	100
NumVersion	Single	4
Descripcion	Memo	-

Usuarios

ID_Usuario	Long Integer	4
Fecha	Date/Time	8
Nombre	Text	100
ApePaterno	Text	100
ApeMaterno	Text	100
CalleNumero	Text	100
Colonia	Text	100
CP	Text	10

Edo	Text	100
Pais	Text	100
Lada	Text	15
Telefono	Text	50
Extension	Text	10
Email	Text	100
Notas	Memo	-
Descargado	Yes/No	1

Consultas Almacenadas

C_LicsCD

Nombre del Campo	Tipo	Tamaño
ID_Lic	Long Integer	4
CodCD	Long Integer	4
KeyTipoLic	Long Integer	4
Producto	Text	50
FechaAsignada	Date/Time	8
Concedida	Yes/No	1
FechaConcedida	Date/Time	8
EsReposicion	Yes/No	1

CDPaquete

ID_CD	Long Integer	4
CodCD	Long Integer	4
NoSerie	Text	20
NoVersion	Single	4
ID_Usuario	Long Integer	4
ID_Paquete	Long Integer	4
Paquete	Text	100
Descripcion	Memo	N/A

C_CDUsr

ID_CD	Long Integer	4
ID_Usuario	Long Integer	4
CodCD	Long Integer	4
Fecha	Date/Time	8
Nombre	Text	100
ApePaterno	Text	100
ApeMaterno	Text	100
CalleNumero	Text	100
Colonia	Text	100
CP	Text	10
Cd	Text	100
Edo	Text	100
Pais	Text	100
Lada	Text	15
Telefono	Text	50
Extension	Text	10
Email	Text	100
Notas	Memo	N/A
Descargado	Yes/No	1

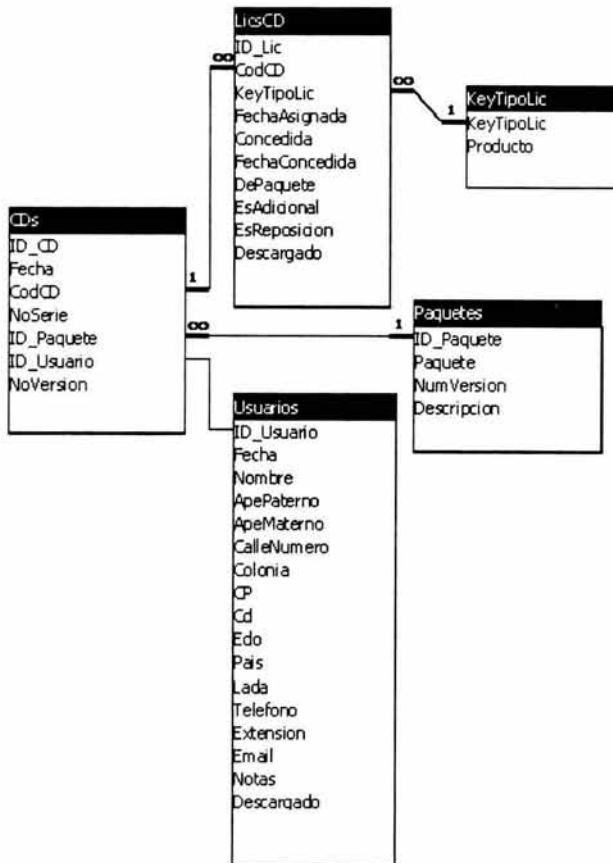


Figura III.8 Diagrama entidad-relación de la base de datos de Internet

Como pudo observarse a lo largo de este capítulo, el sistema esta conformado por 3 subsistemas, 1 sitio WEB, dos bases de datos: una base de datos local y otra alojada en el sitio en Internet. Cada una de ellas es parte fundamental para el buen desempeño del sistema, ya que como podrá apreciarse en el siguiente capítulo su desarrollo es primordial para alcanzar los objetivos y requerimientos del mismo.

En el siguiente capítulo se procede con la fase de desarrollo de sistema, especificando a detalle el desarrollo de cada subsistema que conforma al sistema global.

Desarrollo del sistema

Herramientas para el desarrollo del sistema

Diagramas a bloques

Subsistema Generador de licencias

Subsistema Asistente de registro y activación de software

Servidor de licencias de uso de software en Internet

Subsistema de registro, obtención y consulta de licencia de uso en Internet

En el capítulo anterior se determinaron las características y requerimientos que el sistema debe cumplir al término de su desarrollo, de acuerdo a esto en este capítulo se comienza con el diseño y desarrollo del mismo determinando en primer lugar las herramientas a utilizar, los esquemas de interacción entre los subsistemas que integran al sistema, así como el desarrollo de cada uno de ellos.

Herramientas para diseño y desarrollo del sistema.

• Sistema Operativo Windows

El sistema operativo es el componente de software que en una computadora, administra la utilización de los recursos disponibles y provee la base sobre la cual operan los diversos servicios al usuario, desde utilerías para la administración del sistema hasta las aplicaciones más complejas.

El sistema del presente proyecto de tesis esta enfocado hacia aplicaciones con **plataforma Windows®** de Microsoft, específicamente Windows® 95 o superiores. Las razones por las que fue seleccionada esta plataforma son las siguientes:

- Los sistemas operativos Microsoft desde MS-DOS hasta las diversas versiones de Windows® (3.x, 95, 98, NT, 2000, XP) se han convertido en un estándar, se encuentran ya instalados en la gran mayoría de las máquinas que se encuentran trabajando en México. Siendo más notable en las computadoras de escritorio.
- Al utilizar productos fundamentados en los sistemas operativos de Microsoft, el tamaño de la curva de aprendizaje de la gente que ya lo utiliza, disminuye al no tener que aprender el uso de un sistema operativo nuevo y de herramientas nuevas.
- Las herramientas de Microsoft son soportadas por una gran cantidad de compañías de todos tamaños a nivel Nacional e Internacional.
- Microsoft ha estado mejorando mucho sus productos, para que tengan un mejor desempeño en ambientes de red.

• Bases de Datos Relacionales en Access 2000 de Microsoft

El modelo de bases de datos relacional es el más empleado, ya que todos los datos visibles al usuario están organizados estrictamente como tablas de valores.

Para la creación de las bases de datos que emplea el sistema se eligió el manejador de bases de datos comercial Access de Microsoft tomando en cuenta las siguientes consideraciones:

- Access de Microsoft nos brinda la posibilidad de crear bases de datos relacionales.
 - Satisface las necesidades de crecimiento del sistema, al tener la capacidad para almacenar 2 Giga Bytes de información.
 - Posee un nivel alto y confiable en cuanto a la integridad de los datos.
 - Es compatible con plataforma Windows® de Microsoft.
 - La expectativa de vida, de este manejador, en el mercado es alta ya que la empresa que lo desarrolla genera nuevas actualizaciones.
 - Es compatible con el lenguaje de programación en el que se programará el sistema.
 - Cuenta con sistema de seguridad a nivel de usuario.
 - Se cuenta con versión de Access 2000 por lo que no se tendría que invertir en su adquisición.
 - Otro factor importante es que se posee amplio conocimiento sobre este manejador de bases de datos.
- **Visual Basic 6.0**

Se ha optado por Visual Basic como lenguaje de programación del sistema, debido a su versatilidad para el desarrollo de aplicaciones visuales para Windows. Permite una amplia interacción para diferentes componentes de acceso a datos y la posibilidad de crear componentes ActiveX. Visual Basic contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows®.

Es muy potente permitiendo de manera rápida y sencilla crear aplicaciones para Microsoft Windows®. Proporciona un juego completo de herramientas que facilitan el desarrollo de aplicaciones. Las capacidades de Internet facilitan el acceso a documentos y aplicaciones a través de Internet o intranet desde una aplicación, o la creación de aplicaciones de servidor para Internet.

Para la elección de este lenguaje de programación se tomaron en consideración los siguientes factores:

- Costo
- Lenguaje compatible con plataforma Windows
- Portabilidad
- Incluye herramientas de depuración y creación de formas o pantallas.
- Soporte de Base de Datos Relacionales y, por consiguiente, del lenguaje estructurado de consulta SQL.
- Permite la creación de programas ejecutables.
- Un factor importante a considerar fue el conocimiento que se tiene del lenguaje Visual Basic para el desarrollo del sistema a realizar.

Existen en el mercado otros lenguajes de programación, sin embargo, no todos cumplieron con los requerimientos para el desarrollo de este sistema, a continuación se hace mención de algunos de ellos y las razones por las que no fueron los más óptimos:

DBase:	No soporta programación para entorno gráfico.
Clíper:	No soporta Base de datos relacionales, carece de entorno de programación a la altura de sus competidores.
Visual Fox Pro:	No existen muchas aplicaciones todavía en empresas que lo respalden.
Delphi:	Es una de las opciones que contaron con todo lo requerido, su costo es menor al de otros, cuenta con una rápida y creciente popularidad pero finalmente habría que invertir en su adquisición y en la capacitación para la utilización del mismo.

- **ADO (Objetos de Acceso a Datos)**

Los Objetos de Acceso a Datos constituyen la interfaz de programación de aplicaciones (API) estratégica para datos e información. ADO proporciona un acceso a los datos consistente y de alto rendimiento y es compatible con una variedad de necesidades de programación, incluyendo la creación de clientes para vistas de las bases de datos y de objetos de negocios de la capa intermedia que utilicen aplicaciones, herramientas, lenguajes o exploradores de Internet. ADO se ha diseñado para que sea una de las interfases necesarias para los desarrollos de soluciones orientadas a datos de tipo cliente/servidor⁴ y basadas en Web. Las principales ventajas de ADO son la facilidad de uso, la alta velocidad, el escaso consumo de memoria y la pequeña ocupación de disco.

- **SQL (Lenguaje de consultas estructurado)**

Lenguaje utilizado para consultar, actualizar y administrar bases de datos relacionales. El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, 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.

SQL permite gestionar grupos de registros como si se tratara de una entidad individual, lo que le convierte en un lenguaje adecuado para crear consultas de cierta complejidad.

⁴ **Cliente/Servidor:** el cliente es la computadora que se encarga de efectuar una petición o solicitar servicio, el servidor evalúa la petición del cliente y decide aceptarla o rechazarla consecuentemente. Una vez que el servidor acepta el pedido la información requerida es suministrada al cliente que efectuó la petición, siendo este último el responsable de proporcionar los datos al usuario con el formato adecuado

Este lenguaje ha sido estandarizado y, en la actualidad, la mayoría de los manejadores de bases de datos y el propio ADO aceptan su dialecto. Cuenta con dos categorías distintas de instrucciones: el lenguaje de definición de datos (DDL) y el lenguaje de manipulación de datos (DML). El subconjunto DDL incluye un grupo de instrucciones que permiten crear estructuras de las bases de datos, tales como tablas, campos, índices, etc. El subconjunto DML incluye todos los mandatos que le permiten realizar consultas y modificar los datos contenidos en la base de datos, agregar nuevos registros o borrar los ya existentes.

- **HTML (Hypertext Markup Language)**

HTML es un sistema para marcar o etiquetar un documento de forma que se pueda publicar en el World Wide Web⁵. Los documentos preparados en HTML incluyen gráficos de referencia y etiquetas de formato. Se usa un explorador de Web para ver estos documentos.

El lenguaje HTML es un lenguaje interpretado que permite dar indicaciones precisas al programa cliente, en este caso los navegadores o browsers de cómo debe presentarse el documento en pantalla. Un documento de hipertexto no se compone únicamente de texto, pues también contiene relaciones con otros documentos. Con el paso del tiempo este concepto se ha ampliado aún más, haciendo que los enlaces no solo sean de texto, sino que se complementen con información en otros formatos, como gráficos, sonidos, vídeo, etc. El resultado es un documento que combina muchos elementos multimedia y que permite su difusión por medio del WEB.

- **VBScript (Visual Basic, Scripting Edition)**

El sistema de programación de Visual Basic Scripting Edition (VBScript) es un lenguaje de secuencias de comandos ampliamente difundido y un subconjunto del lenguaje Visual Basic. VBScript está diseñado para incorporarlo a un explorador de Internet, como Microsoft Internet Explorer u otros exploradores. VBScript es un motor de lenguaje ligero y extremadamente rápido diseñado específicamente para entornos como Internet, redes internas (intranets) o el World Wide Web. Tiene la eficacia de Visual Basic y permite que los programadores aprovechen sus conocimientos de Visual Basic para crear rápidamente soluciones para Internet o para el World Wide Web. Acepta un subconjunto de la sintaxis del lenguaje Visual Basic para aplicaciones.

⁵ **World Wide Web:** Es un conjunto de servicios por los que se accede a toda la información de internet, Dicha información puede ser texto, imágenes, video etc.

- **ASP (Active Server Pages)**

ASP es una tecnología de páginas activas que permite el uso de diferentes Scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente.

"Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, Scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el Web". El principio de la tecnología ASP es el **VBScript**, pero existe otra diversidad de lenguajes de programación que pueden ser utilizados como lo es Perl, JScript, etc.

ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del Script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del WEB, es ejecutado, por lo que los usuarios no tienen acceso más que a la página resultante en su navegador.

Para explicar mejor el funcionamiento del ASP se presenta la siguiente gráfica:



Figura IV.1 Funcionamiento de ASP

Entre sus funciones principales están, el acceso a Base de Datos, envío de correo electrónico, creación dinámica de gráficos y otros.

Finalmente, de acuerdo al análisis realizado en el capítulo anterior y ya determinadas las herramientas a emplear, se procede con el diseño y desarrollo del sistema propuesto.

Por lo que se ha ido analizando a lo largo del sistema, se puede determinar que el sistema queda integrado por tres subsistemas y un servidor, que son:

- Subsistema Generador y administrador de licencias de uso de software
- Subsistema Asistente de activación de software
- Subsistema de registro, obtención y consulta de licencia de uso en Internet
- Servidor de licencias de uso en Internet

Diagramas a bloques

La siguiente figura muestra el diagrama a bloques del sistema final, conteniendo la interacción entre los subsistemas, el servidor y su ubicación física, es decir, si esta instalado en alguna computadora de la empresa desarrolladora del software de aplicación protegido, en el sitio de Internet de la misma ó en la computadora del usuario final del programa de aplicación protegido.

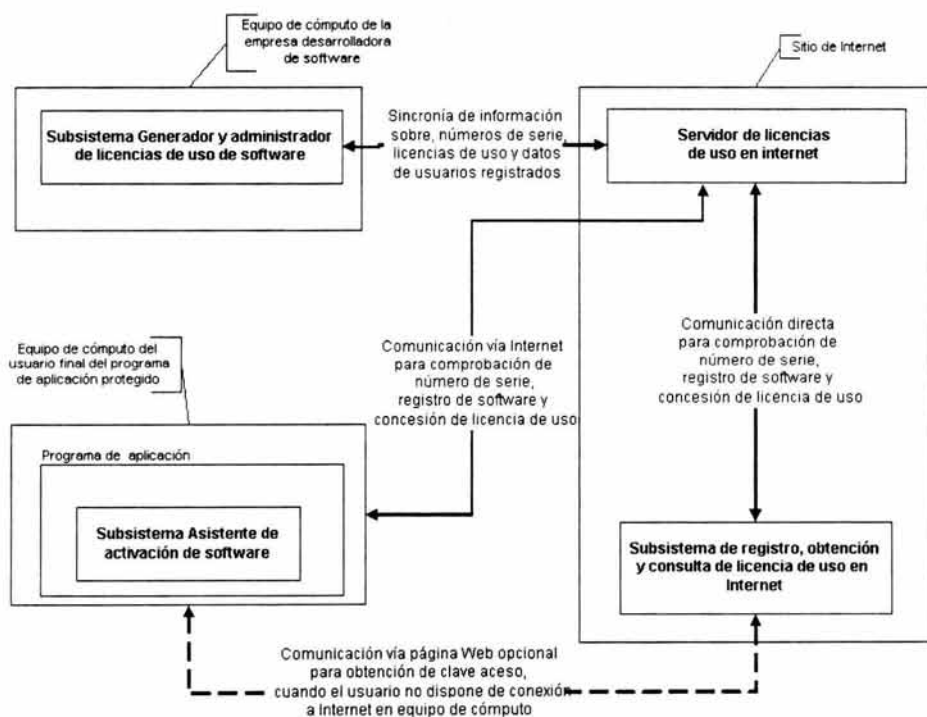


Figura IV.2 Esquema global del sistema

Interacción entre el Subsistema generador y administrador de licencias de uso y el servidor de licencias de uso en Internet:

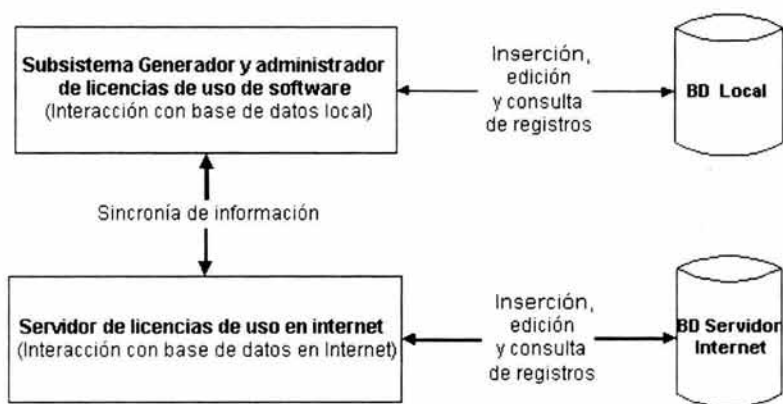


Figura IV.3 Subsistema Generador y Servidor de Licencias.

Interacción entre el Subsistema asistente de activación de software y el servidor de licencias de uso en Internet:

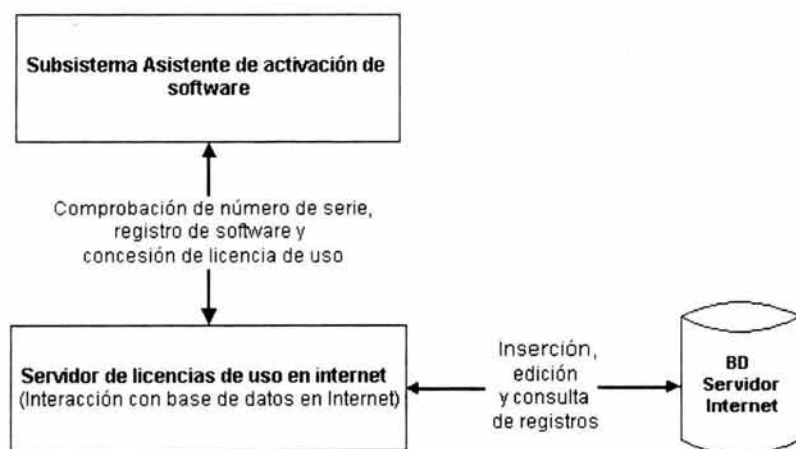


Figura IV.4 Subsistema asistente de activación de software y el servidor de licencias

Interacción entre el Servidor de licencias de uso en Internet y los subsistemas:

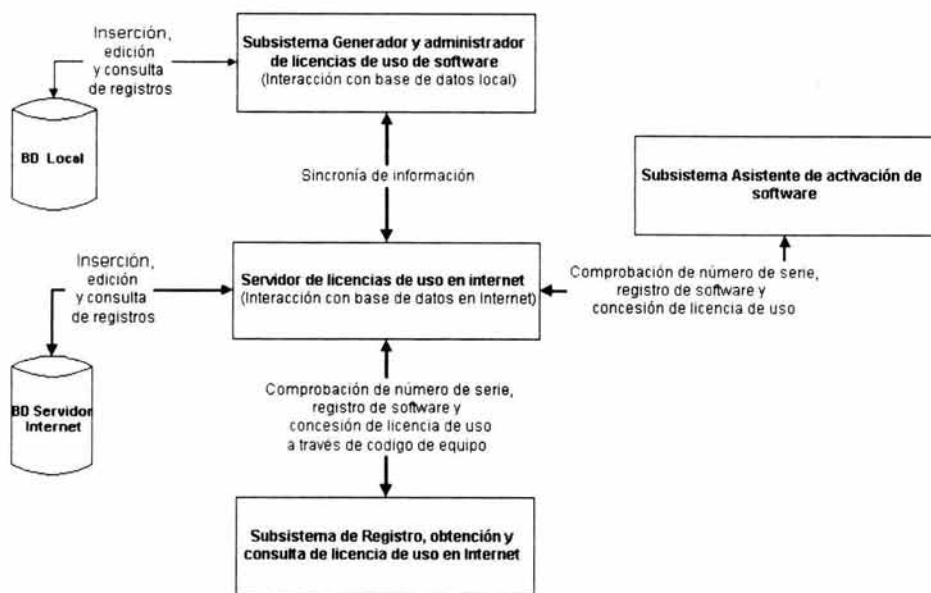


Figura IV.5 Servidor de licencias de uso en Internet

Interacción entre el Subsistema de registro, consulta y obtención de licencias en Internet y el servidor de licencias de uso en Internet:

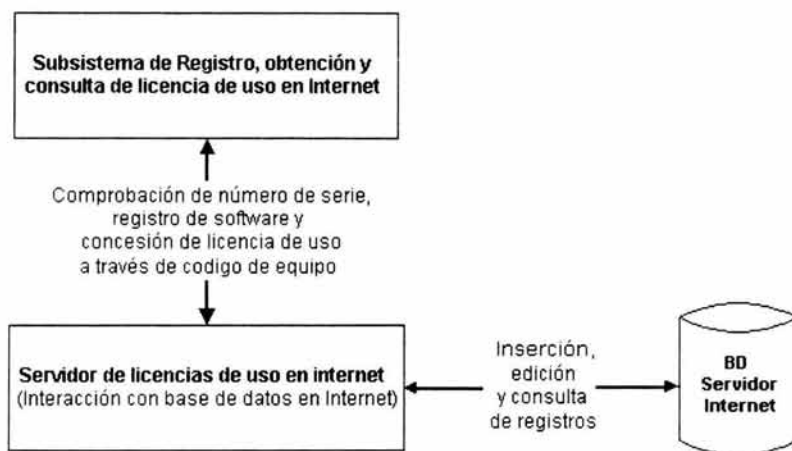


Figura IV.6 Subsistema de registro y el servidor de licencias

Una vez definida la interacción entre los subsistemas se procede a continuación al diseño y programación de los mismos.

Subsistema Generador de licencias.

El siguiente subsistema, como se estableció en el capítulo anterior, es para uso exclusivo de la empresa desarrolladora de la aplicación a proteger contra la piratería. Su función principal es llevar un control de todos y cada uno de los productos de software que la empresa desee proteger, así como también el distribuidor asignado a cada uno de los productos que han salido de la empresa, usuario y licencias por producto. En el desarrollo de este subsistema se utilizaron las siguientes herramientas:

Visual Basic 6: Con esta herramienta se diseña la interfase gráfica (pantallas) de cada uno de los módulos que lo integran así como la programación de estos para su funcionamiento.

Access 2000: La utilidad principal de esta herramienta para la creación y estructuración de la base de datos relacional; conteniendo las tablas, consultas almacenadas y relaciones.

ASP: A través de páginas ASP se establece la comunicación con la base de datos que se encuentra alojada en el sitio en Internet.

Para iniciar el desarrollo de este subsistema y de acuerdo a la información obtenida en el capítulo anterior se comienza por definir los módulos que integran a este subsistema:

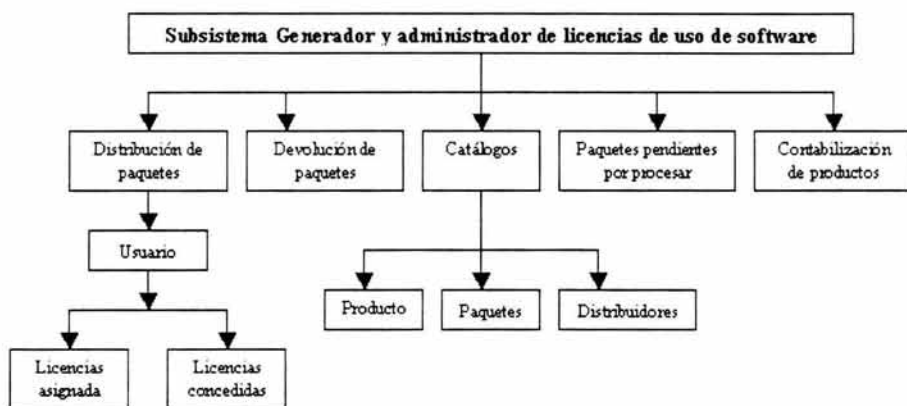


Figura IV.7 Módulos de subsistema generador y admin. de licencias de uso de software.

Es importante señalar que algunos de estos módulos están interactuando con la base de datos ubicada en un sitio en Internet por lo que a lo largo del desarrollo de estos módulos se crearon algunas funciones importantes como son: Función **VerificaCnxInternet** encargada de verificar si hay conexión a Internet para realizar altas, bajas o cambios en la información. Para comprender mejor su funcionamiento se presenta a continuación su código fuente:

```
Public Declare Function InternetGetConnectedStateEx Lib "wininet.dll" Alias
    "InternetGetConnectedStateExA" (ByRef IpdwFlags As Long, ByVal IpszConnectionName As String,
    ByVal dwNameLen As Long, ByVal dwReserved As Long) As Long

Public Enum EIGCInternetConnectionState
    INTERNET_CONNECTION_MODEM = &H1&
    INTERNET_CONNECTION_LAN = &H2&
    INTERNET_CONNECTION_PROXY = &H4&
    INTERNET_RAS_INSTALLED = &H10&
    INTERNET_CONNECTION_OFFLINE = &H20&
    INTERNET_CONNECTION_CONFIGURED = &H40&
End Enum

Public Function VerificaCnxInternet(Optional ByRef eConnectionInfo As EIGCInternetConnectionState, _
    Optional ByRef sConnectionName As String) As Boolean

    Dim dwFlags As Long, sNameBuf As String, IR As Long, iPos As Long
    sNameBuf = String$(513, 0)
    IR = InternetGetConnectedStateEx(dwFlags, sNameBuf, 512, 0&)
    eConnectionInfo = dwFlags
    iPos = InStr(sNameBuf, vbNullChar)

    If iPos > 0 Then
        sConnectionName = Left$(sNameBuf, iPos - 1)
    Elseif Not sNameBuf = String$(513, 0) Then
        sConnectionName = sNameBuf
    End If
    VerificaCnxInternet = (IR = 1)
End Function
```

Como puede apreciarse en la función anterior, se utiliza el archivo "wininet.dll" que pertenece al sistema operativo y devuelve información referente al estado de la conexión a Internet, incluyendo información sobre la configuración actual del medio de acceso como puede ser a través de módem, red de área local o servidor de acceso a Internet (PROXY). De esta forma el sistema determina rápidamente si se puede procesar información en la base de datos que se encuentra en el sitio en Internet.

Es importante mencionar que para realizar las operaciones de Altas, bajas y cambios se utiliza una función que se encarga de enviar la información correspondiente a la base de datos que se encuentra alojada en el sitio en Internet, cabe mencionar que se utilizan varias de estas funciones aunque sólo cambia el nombre de la función y el nombre de las páginas ".ASP" que realizan la conexión.

El siguiente código fuente muestra la función que se encarga de enviar los tipos de licencia de uso hacia el servidor de licencias:

```
Public Function WebKeyTipoLic(ByVal KeyTipoLic As Long, Optional ByVal Producto As String, Optional Eliminar As Boolean) As Boolean
On Error GoTo falla
Dim PagDestino As String, PagFinal As String, CnxHecha As Boolean

PagDestino = "gl_publicakeytipolic.asp?KeyTipoLic=" & KeyTipoLic &
"&Producto=" & Producto & "&Eliminar=" & IIf(Eliminar, 1, 0)

Load frmIniCnxInternet
FrmIniCnxInternet.URLBase = URLBase
frmIniCnxInternet.PaginaDestino = PagDestino
frmIniCnxInternet.TituloLbl = "Publicando Producto"
frmIniCnxInternet.cmdCancel.Visible = False
frmIniCnxInternet.Show vbModal
CnxHecha = frmIniCnxInternet.CnxEstablecida
If CnxHecha Then
    PagFinal = frmIniCnxInternet.WebBrow.LocationURL
    If InStr(1, PagFinal, "gl_infvalida.htm", vbTextCompare) > 0 Then
        If InStr(1, PagFinal, "?publicacioncompletada=si", vbTextCompare) > 0 Then
            WebKeyTipoLic = True
        ElseIf InStr(1, PagFinal, "?eliminacioncompletada=si", vbTextCompare) > 0 then
            WebKeyTipoLic = True
        Else
            MsgBox "No se completo la publicación." & vbCrLf, vbCritical
        End If
    Else
        MsgBox "Error al publicar." & vbCrLf, vbCritical
    End If
Else
    MsgBox "No se pudo establecer conexión a Internet." & vbCrLf, vbCritical
End If
Set frmIniCnxInternet = Nothing

Exit Function
falla:
End Function
```

En este caso la página `gl_publicakeytipolic.asp` es la página que se utiliza de intermediaria para enviar los datos a la base de datos en Internet, estos son los que se encuentran después del '?' y están asignados a una variable que es recibida por la página ".ASP", es decir : suponiendo que `KeyTipoLic` tiene el valor de 1, producto el valor de "Nuevo producto" y Eliminar un valor de 0, la sentencia queda representada de la siguiente manera: `gl_publicakeytipolic.asp? KeyTipoLic=1&Producto=Nuevo producto&Eliminar=0`

Módulos que integran el subsistema.

- **Módulo de distribución de paquetes de software.**

Este módulo es el más importante dentro de este subsistema ya que aquí se genera el número de inventario, número de serie y etiquetas para el software a proteger contra la piratería.

Dicho módulo consta de una ventana que solicita la fecha que muestra el software de la empresa desarrolladora de la aplicación, número de inventario para su identificación, paquete al que pertenece dicho software, el cual, al realizar la selección automáticamente indica el número de versión del software, a su vez, en dicha ventana, debe especificar el distribuidor al que es enviado.

La salida puede ser por un sólo software o bien por más de uno, para ello solo se tiene que especificar el número de inventario inicial y el número de inventario final, los cuales deben ser números consecutivos. Una vez definida esta información se procede a guardar la información, es decir, la ventana cuenta con un botón guardar/publicar que se encarga de guardar la información en la base de datos local y a su vez se encarga de enviarla a la base de datos que se encuentra en Internet. En este proceso de guardar/publicar se asigna internamente el número de serie correspondiente a dicho paquete, de esta forma una vez almacenado, lo único que resta es imprimir la etiqueta que debe anexarse al empaque del software para que de esta forma el usuario de esta aplicación pueda realizar la activación del mismo.

Tabla correspondiente de la base de datos:

Tabla: CDs		
Campo	Tipo	Descripción
CodCD	Entero	ID de identificación único para cada software.
NumInventario	Entero	Número de inventario para el software
NumSerie	Texto	Número de serie único de 20 dígitos para activación del software
FechaSalida	Date	Fecha de salida de la empresa del software.
FechaPublicado	Date	Fecha en la que se ha enviado la información a la base de datos en Internet.
ID_CDPaquete	Entero	Número que identifica a que paquete pertenece el software.
ID_CDDistribuidor	Entero	Número que identifica a que distribuidor se proporciono el software
ID_Usuario	Entero	Número que permite saber a que usuario final pertenece el software.
Nota	Texto	Permite almacenar alguna anotación de importancia a cerca del software o usuario.

EsDevolución	Boleano	Bandera que identifica si el software fue devuelto por el distribuidor o usuario final.
Publicado	Boleano	Bandera que determina si el software ha sido guardado en la base de datos en Internet.
PorEliminar	Boleano	Bandera que determina si el software ha sido eliminado de la base de datos en Internet.

Diseño Grafico (pantalla):

La interfase gráfica de este módulo se muestra en la siguiente figura:

Figura IV.8 Vista salida de software

Pseudolenguaje:

Guardar y publicar.

1. Valida que se halla introducido una fecha valida para la salida.
2. Valida que el rango de número de inventario inicial y final sea consecutivo en caso de tratarse de más de una salida.
3. Verifica que no exista un número de inventario igual, en caso de encontrarlo envía mensaje y finaliza el proceso o bien continua si la salida es por más de un software.
4. Valida que se halla seleccionado paquete y distribuidor para la salida del software.
5. Si el proceso hasta este momento ha sido satisfactorio se procede a guardar el nuevo registro en la base de datos para el nuevo software.
6. Genera número de serie para el paquete de salida (el algoritmo para la generación de este número es descrita más adelante)

7. Actualiza el registro generado asignándole el número de serie correspondiente.
8. Verifica a que paquete corresponde y genera la licencia de uso correspondiente a cada software que integra dicho paquete.
9. Una vez guardado en la base de datos local, verifica con la función "VerificaCnxInternet" que halla conexión a Internet.
10. En caso de realizar dicha conexión, se envían los datos a guardar en la base de datos en Internet a la función WebNumSerie, la cual mediante la página gl_salidapaquete.asp envía al servidor de licencias en Internet el nuevo registro, a su vez, se envía la licencia de uso disponible para el software mediante la función "WebLicSalidapaquete", la cual, se encarga de enviar esta información mediante la página gl_licsalidasoftware.asp.
11. Ya creado el registro en la base de datos en Internet se actualiza en la base de datos local el campo Publicado asignándole un 1.
12. Finaliza la operación del registro.

En caso de no lograr conexión a Internet las operaciones sólo se efectúan en la base de datos local y al registro Publicado se le asigna el valor de 0, esto permite que se pueda realizar la operación cuando se tenga conexión a Internet.

Algoritmo para generación de número de serie

El objetivo principal del número de serie es identificar de manera única cada paquete de software, además es un medio a través del cual se pueda colocar información oculta que es utilizada para comprobar su autenticidad. El número de serie es generado mediante un algoritmo que realiza la codificación ⁶ de un número entero aleatorio y se añade un código que permita verificar su validez, es decir, que la mínima alteración lo invalide.

A cada paquete de software se le genera un número de serie que lo identifica de manera única. Así mismo cada paquete podría contar con un número de inventario independientemente del formato que se desee utilizar para identificar cada paquete producido.

El número de serie esta conformado por cuatro series de cinco caracteres alfanuméricos, como se muestra a continuación:

XXXXX - XXXXX - XXXXX - XXXXX

Donde la "X" puede ser cualquier dígito del cero al nueve o una letra del alfabeto, es decir, cualquier letra entre la "A" y la "Z", exceptuando la letra "Ñ". Considerando que cada carácter del número de serie puede tomar 35 diferentes

⁶ **Codificar:** Consiste en transformar un mensaje inteligible en otro que no lo es, mediante la utilización de claves que solo el emisor y receptor conocen.

valores y son 20 caracteres los que conforman al número de serie, por lo tanto se tienen 35^{20} posibles combinaciones distintas. Un ejemplo real podría ser:

ZTDH8 - N9RZH - B73CZ - RIV2L

Se han utilizado caracteres alfabéticos para la representación de cantidades mayores a nueve, con lo cual pueden representarse números del 10 al 35, es a través de valores numéricos como se codifica la información dentro del número de serie; a continuación se muestra una tabla que describe la asignación de una letra a los valores numéricos mayores a 9.

10 = A	15 = F	20 = K	25 = P	30 = U	35 = Z
11 = B	16 = G	21 = L	26 = Q	31 = V	
12 = C	17 = H	22 = M	27 = R	32 = W	
13 = D	18 = I	23 = N	28 = S	33 = X	
14 = E	19 = J	24 = O	29 = T	34 = Y	

Como puede observarse en la tabla anterior se ha asignado una letra a un número en específico, esto con la finalidad de poder utilizar letras para representar valores numéricos mayores a nueve en el número de serie. Es importante mencionar que el orden en el que se han asignado los caracteres numéricos puede modificarse para diferenciar los números de serie de una empresa a otra y además proporcionar un grado mayor de complejidad en la decodificación de los mismos, así por ejemplo podría tenerse el siguiente arreglo de asignación:

10 = H	15 = R	20 = A	25 = N	30 = L	35 = M
11 = O	16 = U	21 = W	26 = D	31 = T	
12 = Z	17 = B	22 = V	27 = G	32 = X	
13 = Q	18 = S	23 = C	28 = E	33 = J	
14 = F	19 = Y	24 = I	29 = P	34 = K	

De esta forma no se tiene un sentido lógico de asignación que pudiera permitir a alguien una fácil decodificación de la información oculta dentro del número de serie. Esta asignación aleatoria no es lo único que le brinda seguridad, además, se aplica un algoritmo que permite su comprobación.

Como se ha mencionado con anterioridad cada paquete de software inventariado debe ser registrado en la base de datos del sistema, el cual le asigna un número entero único, este número asignado al número de inventario es generado de forma aleatoria dentro de un intervalo comprendido entre:

-2,147,483,648 y 2,147,483,648

sin permitirse repetir un número ya utilizado. Este número es codificado para así crear el número de serie único, es decir, este número constituye la base para la

generación del número de serie, además de ser el elemento de información oculto que permite comprobar su autenticidad.

A continuación se describe el algoritmo para la creación del número de serie:

- PASO 1: Ajustar el número aleatorio generado por el sistema a 10 dígitos completando con ceros a la izquierda si es necesario.
- PASO 2: Codificar cada uno de los 10 dígitos del número aleatorio, mediante operaciones matemáticas entre si mismos, pero con la restricción de que el resultado de la operación sea un número entero comprendido dentro del intervalo de 0 a 35. Con ello se genera un arreglo de 10 números enteros.
- PASO 3: Generar un segundo arreglo de 10 números donde cada número del arreglo es el resultado de operaciones matemáticas entre los números del primer arreglo, de igual forma con la restricción de que el resultado sea un número entero comprendido dentro del intervalo de 0 a 35. Básicamente este segundo arreglo es el código de verificación del primer arreglo.
- PASO 4: Codificar cada número de ambos arreglos a su correspondiente dígito o letra según sea su valor y de acuerdo a la tabla de asignación a utilizar.
- PASO 5: Finalmente unir cada número ya codificado de ambos arreglos, para conformar los 20 caracteres del número de serie.

A continuación se presenta a manera de ejemplo la creación del número de serie para el número aleatorio -514821463, empleando operaciones matemáticas básicas (suma y resta), aclarando que podrían emplearse otros operadores matemáticos para darle mayor complejidad a la codificación del número de serie resultante (por razones de seguridad, no se muestran las verdaderas operaciones que realiza el sistema).

PASO 1: Ajustar a 10 dígitos número aleatorio: -0514821463

PASO 2: Arreglo 1, a continuación se muestra la tabla de funciones matemáticas que a realizar con los dígitos del número aleatorio. Se le llama arreglo "A" y cada dígito del número aleatorio se representa con la letra D y un número que identifica su posición de izquierda a derecha, de esta forma D1 corresponde con el 0, D2 con el 5, hasta D10 que corresponde al 3.

$$A[1] = D1 + D9 + 11$$

$$A[2] = D2 + D6 + D7 + 5$$

$$A[3] = D3 + D6 + D9 + 7$$

$$A[4] = D4 + D7 + D8 + 8$$

$$A[5] = D5 + D7 + D9 + 3$$

$$A[6] = D6 + D8 + D9 + 2$$

$$A[7] = D7 + D9 + 3$$

$$A[8] = D8 + D8 + D10 + 8$$

$$A[9] = D9 + D10 + 6$$

$$A[10] = D10 + Dsigno + 1$$

“Dsigno” es igual a 10 si el signo del número aleatorio es positivo, de lo contrario es igual a 20. Aplicando las funciones matemáticas se obtienen los siguientes resultados:

$$A[1] = 0 + 6 + 11 = 17$$

$$A[2] = 5 + 8 + 4 + 5 = 22$$

$$A[3] = 1 + 2 + 6 + 7 = 16$$

$$A[4] = 4 + 1 + 4 + 8 = 17$$

$$A[5] = 8 + 1 + 6 + 3 = 17$$

$$A[6] = 2 + 4 + 6 + 2 = 14$$

$$A[7] = 1 + 6 + 3 = 10$$

$$A[8] = 4 + 4 + 6 + 8 = 22$$

$$A[9] = 6 + 3 + 6 = 15$$

$$A[10] = 3 + 20 + 1 = 24$$

PASO 2: Arreglo 2, a continuación se muestra la tabla de funciones matemáticas que a realizarse con los valores numéricos del arreglo “A”; a este segundo arreglo se le llama arreglo “B”.

$$B[1] = A[5] + 5$$

$$B[2] = A[7] + 9$$

$$B[3] = A[1] - 7$$

$$B[4] = A[9] + 11$$

$$B[5] = A[6] + 6$$

$$B[6] = A[2] + 3$$

$$B[7] = A[4] - 4$$

$$B[8] = A[10] + 2$$

$$B[9] = A[8] - 5$$

$$B[10] = A[3] - 4$$

Aplicando las funciones matemáticas se obtienen los siguientes resultados:

$$B[1] = 17 + 5 = 22$$

$$B[2] = 10 + 9 = 19$$

$$B[3] = 17 - 7 = 10$$

$$B[4] = 15 + 11 = 26$$

$$B[5] = 14 + 6 = 20$$

$$B[6] = 22 + 3 = 25$$

$$B[7] = 17 - 4 = 13$$

$$B[8] = 24 + 2 = 26$$

$$B[9] = 22 - 5 = 17$$

$$B[10] = 16 - 4 = 10$$

PASO 4: Se Codifica cada número de ambos arreglos a su correspondiente dígito o letra según sea su valor de acuerdo a la tabla de asignación siguiente:

Tabla de asignación						
10 = H	14 = F	18 = S	22 = V	26 = D	30 = L	34 = K
11 = O	15 = R	19 = Y	23 = C	27 = G	31 = T	35 = M
12 = Z	16 = U	20 = A	24 = I	28 = E	32 = X	
13 = Q	17 = B	21 = W	25 = N	29 = P	33 = J	

Después de codificar ambos arreglos se obtiene:

A[1] = 17 = B	B[1] = 22 = V
A[2] = 22 = V	B[2] = 19 = Y
A[3] = 16 = U	B[3] = 10 = H
A[4] = 17 = B	B[4] = 26 = D
A[5] = 17 = B	B[5] = 20 = A
A[6] = 14 = F	B[6] = 25 = N
A[7] = 10 = H	B[7] = 13 = Q
A[8] = 22 = V	B[8] = 26 = D
A[9] = 15 = R	B[9] = 17 = B
A[10] = 24 = I	B[10] = 10 = H

PASO 5: Unir cada número ya codificado de ambos arreglos, para conformar los 20 caracteres del número de serie.

<i>Distribución de arreglos "A" y "B" para formar el número de serie resultante</i>																			
A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]	B[7]	B[8]	B[9]	B[10]
B	V	U	B	B	F	H	V	R	I	V	Y	H	D	A	N	Q	D	B	H

Finalmente se obtiene que el número de serie correspondiente a el número aleatorio: -0514821463 es: BVUBB - FHVRI - VYHDA - NQDBH. Cualquier alteración en cualquiera de los caracteres que lo conforman lo invalidarían, debido a que no se cumplirían las funciones matemáticas que lo generaron. Por cual es indispensable aplicar las mismas funciones matemáticas tanto en la codificación como en su decodificación.

Se han unido los números ya codificados de ambos arreglos, uno a continuación del otro, bien podrían haberse intercalado de una forma aleatoria para adicionarle un poco más de complejidad; obviamente empleando el mismo orden de intercalación para todos los números de serie, tanto en la codificación como en la decodificación. Es importante mencionar que el algoritmo expuesto no esta limitado a utilizarse para generar números de serie de 20 caracteres o seguir las funciones matemáticas empleadas, lo importante es aplicar la mecánica del algoritmo y aplicar cualquiera de las funciones, obviamente entre más complejas proveen mayor seguridad al número de serie resultante.

• Módulo Usuario de paquete de software

En este módulo se encuentran listados todos los números de serie correspondientes a los productos que han salido de la empresa. Dicho módulo consta de una ventana que muestra una lista en la que se puede seleccionar el número de inventario del producto del que desea obtener información como: paquete al que pertenece, distribuidor al que se otorgo, datos del usuario al que quedo registrado el producto como son Nombre, Domicilio, Teléfono, etc, licencias concedidas y licencias disponibles. Además cuenta con una opción que permite hacer uso nuevamente del producto si este ha sido registrado como de devolución, en dicho caso se puede cambiar el número de serie correspondiente a ese número de inventario. A su vez se permite asignar licencias adicionales o por reposición, en este mismo módulo, estas licencias pueden ser por paquete o bien por producto, es decir si por ejemplo un usuario adquiere un paquete que consta de 3 productos X, y el usuario desea realizar nuevamente la instalación en otro equipo que acaba de adquirir, debe solicitar a la empresa, se le habilite una licencia por reposición del producto que nuevamente desea instalar o bien de todo el paquete en dicho caso.

Tabla correspondiente de la base de datos:

Tabla: C_LicsCD		
Campo	Tipo	Descripción
ID_Lic	Entero	ID de identificación único para cada licencia.
CodCD	Entero	ID de identificación único para cada software.
KeyTipoLic	Entero	ID de identificación único para cada producto.
Producto	Texto	Nombre del producto.
FechaAsignada	Date	Fecha de salida de la empresa del software.
Concedida	Boleano	Bandera que determina si la licencia ya fue activada.
FechaConcedida	Date	Fecha en que el usuario final activo su licencia de uso.
EsReposición	Boleano	Bandera que determina si la licencia es de reposición.
Publicado	Boleano	Bandera que determina si la licencia ha sido guardada en la base de datos en Internet.
PorEliminar	Boleano	Bandera que determina si la licencia ha sido eliminada de la base de datos en Internet.
EsDevolucion	Boleano	Bandera que identifica si el software fue devuelto por el distribuidor o usuario final

Tabla: Usuarios		
ID_Usuario	Entero	ID de identificación único para cada usuario.
Fecha	Texto	Fecha de registro.
Nombre	Texto	Nombre de usuario que registro el producto

ApePaterno	Texto	Dato del usuario que registro el producto
ApeMaterno	Texto	Dato del usuario que registro el producto
CalleNumero	Texto	Dato del usuario que registro el producto
Colonia	Texto	Dato del usuario que registro el producto
CP	Texto	Dato del usuario que registro el producto
Cd	Texto	Dato del usuario que registro el producto
Edo	Texto	Dato del usuario que registro el producto
Pais	Texto	Dato del usuario que registro el producto
Lada	Texto	Dato del usuario que registro el producto
Telefono	Texto	Dato del usuario que registro el producto
Extension	Texto	Dato del usuario que registro el producto
Email	Texto	Dato del usuario que registro el producto
Notas	Memo	Campo de Anotaciones importantes.

Diseño Grafico (pantalla):

La interfase gráfica de este módulo se muestra en la siguiente figura:

Num. Inventario: 1

Fecha de salida: 01/Enero/2003 Hoy

Este CD fue devuelto
Presionar para restituir

Datos de Usuario:

Nombre: Agustín Ocampo Meyer

Calle/Número: Calle 1 Num 456

Col: Del Valle CP: 12345

Cd: Tapachula Edo: Chiapas

Pais: México

Lada: 962 Telefono: 45 88 75 22 Extension: 210

E-mail: meyer@hotmail.com

Paquete: Paquete 1

Producto 1
Producto 2

Versión: 2003

Distribuidor: SM Software, S. A. de CV

Nota:

Agregar Licencias

Licencias Disponibles

Licencia Adicional

Licencias Concedidas

Licencia de Paquete

Figura IV.9 Vista Usuarios

Pseudolenguaje:

Al mostrarse la pantalla anterior se realizan internamente las siguientes operaciones:

1. Llenado de la lista de números de inventario y selección del primero.
2. Recuperación de información del número de inventario seleccionado para su impresión en pantalla (paquete, versión, distribuidor y nota).
3. Se verifica número de licencias disponibles y se listan con fecha de asignación, así como también, se especifica, si la licencia es de paquete, reposición o adicional.
4. Se verifica número de licencias concedidas y se listan con la fecha en que fueron activadas para su uso, así como también, se especifica, si la licencia concedida fue de paquete, reposición o adicional.
5. Recuperación de datos de usuario que registro el producto correspondiente a dicho número de inventario para su impresión en pantalla (nombre, dirección, teléfono, etc).

Cabe mencionar que la información mostrada en pantalla no es modificable, excepto la de los productos de devolución, la cual solo puede ser modificada por otro número de serie o distribuidor. Por otro lado, como puede apreciarse en la figura IV.9, se muestra en pantalla el botón "Agregar licencias", al hacer clic en dicho botón se muestra la siguiente pantalla:

Diseño Grafico (pantalla):

La interfase gráfica para agregar licencias se muestra en la siguiente figura:

La imagen muestra una ventana de software con el título "Licencias". En la parte superior derecha hay un campo "Fecha:" con el valor "01/Junio/2003" y un botón "Hoy". A la izquierda hay un icono de un documento con un signo de prohibido. Debajo del icono, el texto "Licencia:" está seguido de dos botones de radio: "Adicional" (que está seleccionado) y "Reposición". Abajo de esto, hay dos botones de radio: "Paquete:" (seleccionado) y "Producto:". Cada uno de estos botones de radio está seguido de un campo de texto con una flecha hacia abajo, que contiene "Paquete 1" y "Producto 1" respectivamente. En la parte inferior izquierda, hay un campo "Número de Licencias:" con el valor "1". A la derecha de este campo hay un botón "Agregar".

Figura IV.10 Vista Licencia Adicional/Reposición

En esta parte se puede asignar nuevas licencias disponibles para su uso a un producto en específico, ya sea adicional o de reposición.

Pseudolenguaje:**Asigna nueva licencia de uso.**

1. Valida que se halla introducido una fecha válida.
2. Verifica que el número de inventario al que se desea asignar la licencia no sea de devolución.
3. Si el proceso hasta este momento ha sido satisfactorio se procede a guardar el nuevo registro en la base de datos.
4. Una vez guardado en la base de datos local, verifica con la función "VerificaCnxInternet" que halla conexión a Internet.
5. En caso de realizar dicha conexión se envían los datos a guardar en la base de datos en Internet a la función "WebLicAdicRep" en la cual mediante la página `gl_licadirep.asp` enviamos al servidor de licencias en Internet el nuevo registro.
6. Ya creado el registro en la base de datos en Internet se actualiza en la base de datos local el campo Publicado asignándole un 1.
7. Finaliza la operación del registro.

En caso de no lograr conexión a Internet las operaciones sólo se efectúan en la base de datos local y al registro correspondiente se le asigna Publicado = 0, esto permite que se pueda realizar la operación cuando se tenga conexión a Internet.

Al salir de esta ventana se actualiza la información en pantalla, mostrándose la nueva licencia en la lista de licencias disponibles.

Por otro lado en la pantalla de la figura IV.9 se puede realizar la eliminación de licencias, como se explica enseguida.

Eliminar licencia de uso disponible.

Junto al botón "Agregar licencias" de la figura IV.9 se ubica el botón "Elimina licencia", dicho botón se utiliza para eliminar la licencia que se encuentre seleccionada en la lista de licencias disponibles. Las licencias concedidas no pueden ser eliminadas.

Pseudolenguaje:

1. Verifica que se encuentre un elemento seleccionado de la lista de licencias disponibles.
2. Verifica que el número de inventario al que se desea eliminar la licencia no sea de devolución.

3. Verifica que el campo Publicado sea igual a 1, de no ser así se procede a su eliminación inmediata en la base de datos local y finaliza operación.
4. En caso contrario verifica conexión a Internet mediante la función "VerificaCnxInternet".
5. En caso de realizar dicha conexión se envía el ID de licencia a eliminar en la base de datos en Internet a la función "WebEliminarLic" en la cual mediante la página `gl_eliminalicencia.asp` envía al servidor de licencias en Internet el registro.
6. Ya eliminado el registro de la base de datos en Internet se procede a eliminar el dato de la base de datos local.
7. Finaliza la operación del registro.

En caso de no lograr conexión a Internet y el campo Publicado sea igual a "1" la eliminación no se realiza en la base de datos local y se coloca en el registro correspondiente el campo `PorEliminar=1`, esto permite que se pueda realizar la operación cuando se tenga conexión a Internet.

• Módulo Devolución de paquete de software

En este módulo se realiza la devolución de un paquete. Dicho módulo consta de una ventana que solicita la fecha de devolución del paquete y número de inventario correspondiente para su identificación. La devolución puede realizarse por un sólo paquete o bien por más de uno, para ello solo se debe especificar el número de inventario de los paquetes a devolver, los cuales deben ser números consecutivos.

La devolución sólo se puede realizar si el paquete no ha sido registrado aún por ningún usuario, en caso contrario no se puede realizar. Si se realizó correctamente, automáticamente se agrega a la lista de devoluciones especificando número de inventario, paquete al que pertenece, fecha de devolución, distribuidor, etc. Es importante mencionar que el número de inventario junto con las licencias que tenía asignadas no se borran de la base de datos local, solamente se elimina de la base de datos alojada en el sitio en Internet, esto debido a que si el empaque del producto devuelto ya ha sido abierto, este no se pueda activar una vez realizada la devolución.

La finalidad de mantenerlo en la base de datos local es para que pueda volver a ser reutilizado dicho número de inventario, ya sea con las mismas licencias de uso disponibles que tenía y mismo número de serie, o bien por lo contrario con diferentes, pero el número de inventario sigue siendo el mismo.

Tabla correspondiente de la base de datos:

Tabla: CDsDevolucion		
Campo	Tipo	Descripción
ID_CDsDevolucion	Entero	ID de identificación único para cada devolución..
CodCD	Entero	ID de identificación único para cada software.
NumSerie	Texto	Número de serie para activación del software
NumInventario	Entero	Número de inventario para el software
FechaSalida	Date	Fecha de salida de la empresa del software.
FechaDevolucion	Date	Fecha de devolución del software.
ID_CDPaquete	Entero	ID de identificación único para cada paquete.
ID_CDDistribuidor	Entero	ID de identificación único para cada distribuidor.
ID_Usuario	Entero	Número de identificación del usuario final del software.
Licencias	Texto	Lic. de cada producto que conforma el paquete.

Tabla: C_CDDevolucion		
Campo	Tipo	Descripción
NumInventario	Entero	Número de inventario para el software
FechaSalida	Date	Fecha de salida de la empresa del software.
FechaDevolucion	Date	Fecha de devolución del software.
ID_CDPaquete	Entero	ID de identificación único para cada paquete.
CDPaquete	Texto	Nombre asignado al paquete.
NumVersion	Texto	Número de versión del paquete Ej. Paquete 2002.
ID_CDDistribuidor	Entero	ID de identificación único para cada distribuidor.
CDDistribuidor	Texto	Nombre asignado al distribuidor.

Diseño Grafico (pantalla):

La interfase gráfica de este módulo se muestra en la siguiente figura:

Figura IV.11 Vista Devoluciones

Pseudolenguaje:

1. Valida que se halla introducido una fecha válida.
2. Valida que el rango de número de inventario inicial y final sea consecutivo en caso de tratarse de más de una salida
3. Verifica que el número de inventario a devolver no este registrado ya como devolución.
4. Verifica que el número de inventario a devolver no haya sido ya registrado por algún usuario.
5. Si el proceso hasta este momento ha sido satisfactorio se procede a realizar la devolución.
6. Se guarda en la base de datos local un registro del número de inventario que ha sido devuelto (paquete, fecha de devolución, distribuidor, etc).
7. Una vez guardado en la base de datos local, verifica con la función "VerificaCnxInternet" que halla conexión a Internet.
8. En caso de realizar dicha conexión se envían los datos a guardar en la base de datos en Internet a la función WebDevolucion en la cual mediante la página `gl_devolucionpaquete.asp` enviamos al servidor de licencias el número de inventario a devolver. Este proceso consiste en eliminar ese registro de la base de datos alojada en el sitio en internet.
9. Ya realizada la eliminación en la base de datos en Internet se actualiza en la base de datos local el campo `EsDevolucion` asignándole un 1 y `Publicado` con un 0.
10. Finaliza la operación del registro.

En caso de no lograr conexión a Internet las operaciones sólo se efectúan en la base de datos local y al registro `EsDevolucion` se le asigna el valor de 1 y `Publicado` = 1, esto permite que se pueda realizar la operación cuando se tenga conexión a Internet.

CATALOGOS**• Módulo Productos de software.**

En este módulo se lleva un registro de los productos de software que la empresa ha decidido proteger contra la piratería. Dicho módulo consta de una ventana que muestra una lista donde se encuentran listados los productos ya almacenados en la base de datos, en dicha ventana el usuario puede agregar, actualizar o eliminar productos según sus necesidades. Estos productos son almacenados en la base de datos local y a su vez almacenados en la base de datos en Internet, esto debido a que de esa forma se identifica el producto del cual el usuario desea obtener licencia de uso desde Internet.

Tabla correspondiente de la base de datos:

Tabla: KeyTipoLic		
Campo	Tipo	Descripción
KeyTipoLic	Entero	ID de identificación único para cada producto.
Producto	Texto	Nombre asignado al producto.
Publicado	Boleano	Bandera que determina si el producto ha sido guardado en la base de datos en Internet.
PorEliminar	Boleano	Bandera que determina si el producto ha sido eliminado de la base de datos en Internet.

Diseño Grafico (pantalla):

La interfase gráfica de este módulo se muestra en la siguiente figura:

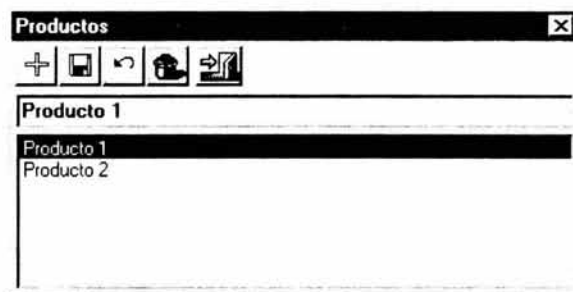


Figura IV.12 Vista productos

Pseudo código:**Guardar o Actualizar Productos.**

1. Verifica si se trata de una actualización o un nuevo producto.
2. Valida que se halla introducido un nombre para el nuevo producto.
3. Verifica que no exista un producto con el mismo nombre.
4. Guarda nuevo registro en la base de datos para el nuevo producto o bien actualiza el registro seleccionado con la nueva información proporcionada.
5. Agrega a la lista el nombre del producto.
6. Una vez guardado en la base de datos local, verifica con la función VerificaCnxInternet que halla conexión a Internet.
7. En caso de realizar dicha conexión se envían los datos a guardar en la base de datos en Internet a la función WebKeyTipoLic en la cual mediante la página gl_publicakeytipolic.asp enviamos al servidor de licencias en Internet el nuevo registro.

8. Ya creado el registro en la base de datos en Internet se actualiza en la base de datos local el campo Publicado asignándole un 1.
9. Finaliza operación del registro.

En caso de no lograr conexión a Internet las operaciones sólo se efectúan en la base de datos local y al registro Publicado se le asigna el valor de 0, esto permite que se pueda realizar la operación cuando se disponga de conexión a Internet.

Eliminar Productos.

1. Verifica que se encuentre seleccionado de la lista un producto.
2. Verifica que el campo Publicado sea igual a 1, de no ser así se procede a su eliminación inmediata en la base de datos local y finaliza operación.
3. Verifica conexión a Internet mediante la función VerificaCnxInternet .
4. En caso de realizar dicha conexión se envía el ID de producto a eliminar en la base de datos en Internet a la función WebKeyTipoLic en la cual mediante la página gl_publicakeytipolic.asp envía al servidor de licencias en Internet el registro.
5. Ya eliminado el registro de la base de datos en Internet se procede a eliminar el dato de la base de datos local.
6. Finaliza la operación del registro.

En caso de no lograr conexión a Internet y el campo Publicado sea igual a "1" la eliminación no se efectúa en la base de datos local y se coloca en el registro PorEliminar el valor de 1, esto permite que se pueda realizar la operación cuando se disponga de conexión a Internet.

• Módulo Paquetes de software.

Una vez que se han creado los productos se pueden elaborar paquetes de acuerdo a las necesidades de la empresa, es decir, si por ejemplo la empresa desarrolla dos programas X, y cada uno de estos se va a proteger contra la piratería, ambos pueden formar parte de un paquete de promoción.

El módulo consta de una ventana que muestra una lista donde se encuentran listados los paquetes ya almacenados en la base de datos, en dicha ventana el usuario (persona encargada de administrar este subsistema dentro de la empresa que implanta este sistema de protección) puede agregar, actualizar o eliminar los paquetes según sus necesidades. Estos paquetes son almacenados en la base de datos local y a su vez almacenados en la base de datos en Internet, esto debido a que de esa forma se identifica a que paquete pertenece el producto del cual el usuario desea obtener licencia de uso.

Tablas correspondientes de la base de datos:

Para este módulo se utilizan dos tablas, **CDPaquete** que almacena los paquetes y **CDDetallePaquete** que almacena los productos correspondientes a cada paquete.

Tabla: CDPaquete		
Campo	Tipo	Descripción
ID_CDPaquete	Entero	ID de identificación único para cada paquete.
CDPaquete	Texto	Nombre asignado al paquete.
NumVersion	Texto	Número de versión del paquete Ej. Paquete 2002.
Descripcion	Texto	Descripción del paquete.
Publicado	Boleano	Bandera que determina si el producto ha sido guardado en la base de datos en Internet.
PorEliminar	Boleano	Bandera que determina si el producto ha sido eliminado de la base de datos en Internet.

Tabla: CDDetallePaquete		
Campo	Tipo	Descripción
ID_CDDetallePaquete	Entero	ID de identificación único para cada paquete.
ID_CDPaquete	Númerico	Nombre asignado al paquete.
KeyTipoLic	Texto	Nombre del producto.

Diseño Grafico (pantalla):

La interfase gráfica de este módulo se muestra en la siguiente figura:

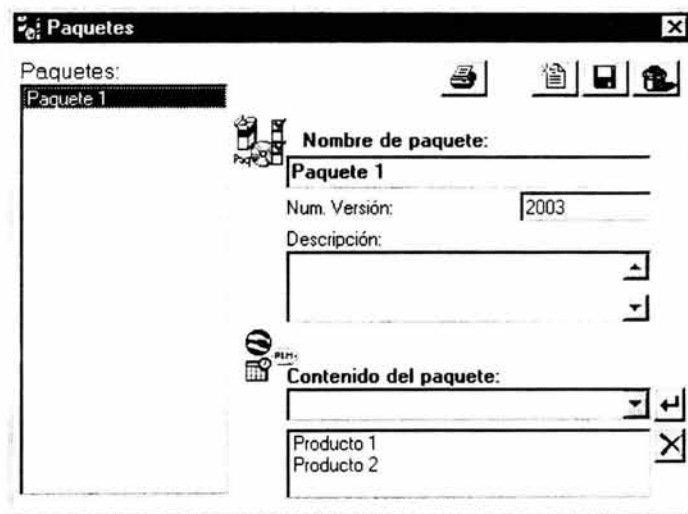


Figura IV.13 Vista paquetes

Pseudolenguaje :

Guardar o Actualizar Paquetes.

1. Verifica si se trata de una actualización o un nuevo paquete.
2. Valida que se halla introducido un nombre para el nuevo paquete.
3. Verifica que no exista un paquete con el mismo nombre.
4. Guarda nuevo registro en la base de datos para el nuevo paquete o bien actualiza el registro seleccionado con la nueva información proporcionada.
5. Agrega a la lista el nombre del paquete.
6. Una vez guardado en la base de datos local, verifica con la función "VerificaCnxInternet" que halla conexión a Internet.
7. En caso de realizar dicha conexión se envían los datos a guardar en la base de datos en Internet a la función WebPaquetes en la cual mediante la página gl_paquete.asp enviamos al servidor de licencias en Internet el nuevo registro.
8. Ya creado el registro en la base de datos en Internet se actualiza en la base de datos local el campo Publicado asignándole un 1.
9. Finaliza operación del registro.
10. Almacenado el registro se habilita la lista de productos, para asignar los productos que corresponden al paquete creado.

En caso de no lograr conexión a Internet las operaciones sólo se efectúan en la base de datos local y al registro Publicado se le asigna el valor de 0, esto permite que se pueda realizar la operación cuando se tenga conexión a Internet.

Eliminar Paquetes.

1. Verifica que se encuentre seleccionado de la lista un paquete, y que este no haya sido ya asignado a alguna salida, ya que para eliminarlo, el paquete no debe estar en uso.
2. Verifica que el campo Publicado sea igual a 1, de no ser así se procede a su eliminación inmediata en la base de datos local y finaliza operación.
3. Verifica conexión a Internet mediante la función VerificaCnxInternet.
4. En caso de realizar dicha conexión se envía el ID de paquete a eliminar en la base de datos en Internet a la función WebPaquetes, la cual mediante la página gl_paquete.asp envía al servidor de licencias en Internet el registro.
5. Ya eliminado el registro de la base de datos en Internet se procede a eliminar el dato de la base de datos local.
6. Finaliza la operación del registro.

En caso de no lograr conexión a Internet y el campo Publicado sea igual a "1" la eliminación no se realiza en la base de datos local y se coloca en el registro PorEliminar el valor de 1, esto permite que se pueda realizar la operación cuando se tenga conexión a Internet.

- **Módulo Distribuidores de software.**

En este módulo se lleva un registro de los distribuidores que se encargan de realizar la labor de venta del software que la empresa ha decidido proteger contra la piratería.

Dicho módulo consta de una ventana que muestra una lista donde se encuentran listados los distribuidores ya almacenados en la base de datos, cada distribuidor cuenta con datos como son: Nombre de la razón social (Distribuidor), RFC, Dirección, Nombre del contacto, Teléfonos, E-Mail, etc. En dicha ventana el usuario puede agregar, actualizar o eliminar distribuidores según sus necesidades.

Debido a que esto solo es de interés para la empresa desarrolladora del software a proteger, los registros son almacenados solo en la base de datos local.

Tabla correspondiente de la base de datos:

Tabla: CDDistribuidor		
Campo	Tipo	Descripción
ID_CDDistribuidor	Entero	ID de identificación unico para cada distribuidor.
CDDistribuidor	Texto	Nombre asignado al distribuidor.
CalleNumero	Texto	Datos del distribuidor
Colonia	Texto	Datos del distribuidor
CP	Texto	Datos del distribuidor
Ciudad	Texto	Datos del distribuidor
Estado	Texto	Datos del distribuidor
Pais	Texto	Datos del distribuidor
RFC	Texto	Datos del distribuidor
Contacto	Texto	Datos del contacto del distribuidor
Puesto	Texto	Datos del contacto del distribuidor
Telefono	Texto	Datos del contacto del distribuidor
Fax	Texto	Datos del contacto del distribuidor
Email	Texto	Datos del contacto del distribuidor
Otros	Texto	Este campo es utilizable en el caso en el que se tenga más de un contacto para el distribuidor.

Diseño Grafico (pantalla):

La interfase gráfica de este módulo se muestra en la siguiente figura:

The screenshot shows a window titled "Distribuidores" with a list on the left containing "SM Software...". The main area contains a form for a distributor with the following fields:

- Distribuidor:** SM Software, S. A. de C.V.
- R.F.C.:** SMS155550-S
- Dirección:**
 - Calle/Num.:** Tacuba 39 2° Piso
 - Colonia:** Juarez **C.P.:** 15862
 - Estado:** DF **Cd.:** México
 - Pais:** México
- Contacto:**
 - Contacto:** Julio Ordoñez Solís
 - Puesto:** Gerente de Ventas
 - Teléfono:** 4521-3158 **Fax:** 1546-5236
 - Email:** julio@sms.com.mx
 - Otros:** (empty field with up/down arrows)

Figura IV.14 Vista distribuidores

Pseudolenguaje:**Guardar o Actualizar Distribuidores.**

1. Verifica si se trata de una actualización o un nuevo distribuidor.
2. Valida que se halla introducido un nombre para el nuevo distribuidor.
3. Guarda nuevo registro en la base de datos para el nuevo distribuidor o bien actualiza el registro seleccionado con la nueva información proporcionada.
4. Agrega a la lista el nombre del distribuidor.
5. Finaliza operación del registro.

Eliminar Distribuidor.

1. Verifica que se encuentre seleccionado de la lista el distribuidor.
2. Si el distribuidor no se encuentra en uso se procede a la eliminación del registro.
3. Finaliza la operación del registro.

- **Módulo registros pendientes por procesar en la base de datos alojada en el sitio en Internet.**

Este módulo se encarga de listar todos aquellos registros que no se han procesado en la base de datos alojada en el sitio en Internet. Es decir, si por ejemplo, al querer registrar alguna salida de software, producto, paquete, etc; no hay conexión a Internet en ese momento, esos registros se quedan pendientes, dichos registros se muestran listados en su correspondiente columna, como se muestra en la siguiente figura.

Si al iniciar o finalizar el subsistema se detecta que existen registros por procesar, inmediatamente manda un mensaje que informa que existen registros por procesar y aparece la pantalla de pendientes. La pantalla esta dividida en dos secciones una para pendientes por guardar y otra para pendientes por eliminar.

Diseño Grafico (pantalla):

La interfase gráfica de este módulo se muestra en la siguiente figura:

Pendientes Fecha de publicación: 22/Noviembre/2002

Los siguientes registros no se han enviado a la base de datos en internet.

Inventario:	Licencia Adic. / Rep:	Producto:	Paquete:
	Producto 1 Producto 2 Producto 1	Producto 1 Producto 2	Paquete 1

Los siguientes registros de eliminación no se han eliminado de la base de datos en internet:

Inventario:	Licencia Adic. / Rep:	Devolución:	Producto:	Paquete:

Figura IV.15 Vista Pendientes

Pseudolenguaje:

1. Verifica que el campo Publicado=0, PorEliminar=1 y EsDevolución=1 de todos los registros que se encuentran en la base de datos local, esto para las tablas que correspondan a registros de números de inventario, licencias, productos y paquetes.
2. Agrega a la lista correspondiente los registros pendientes encontrados.
3. Para procesar dichos registros envía la información a la función correspondiente a cada sección, como por ejemplo "WebNumSerie", "WebLicAdicRep", "WebKeyTipoLic", etc, mencionadas anteriormente.
4. Dichas funciones se encargan de realizar las verificaciones pertinentes para cada sección.

En caso de no poder realizar la conexión a Internet los registros siguen pendientes para su proceso.

- **Módulo contabilización de productos.**

Este módulo consta básicamente de 3 listas que se encargan de mostrar la información de los paquetes que han salido de la empresa, licencias concedidas por paquete y paquetes por distribuidor, es decir, en la pantalla de la figura IV.16 se muestra un ejemplo, en la lista "Cds por paquete" se puede apreciar que existe un paquete de nombre "paquete 1" el cual consta de 2 productos (producto1 y producto 2) del cual existen un total de 4 paquetes de ese tipo fuera de la empresa, en la siguiente lista "Licencias concedidas por paquete", se encuentra el paquete de nombre "paquete 1" del cual, el usuario final que adquirió ese paquete ha hecho uso de la licencia para el producto 1 y para el producto 2, en la tercera y última lista se muestra que el distribuidor con nombre "Distribuidor 1" tiene a su cargo la venta de 4 paquetes. En resumen sólo han salido de la empresa desarrolladora de la aplicación a proteger 4 paquetes, de los cuales solo uno ha sido registrado por un usuario y 3 no han sido registrados (debido a que el distribuidor aún no los a vendido o bien el usuario final que lo adquirió no lo ha registrado aún).

Contabilización			
Total de CDs:	4	CDs sin licencia concedida:	3
		Total Usuarios registrados:	1
Cds por Paquete:	Licencias concedidas por paquete:	CDs por Distribuidor:	Total
<input type="checkbox"/> Paquete 1 TOTAL: 4 <input type="checkbox"/> Producto 1 <input type="checkbox"/> Producto 2	<input type="checkbox"/> Paquete 1 <input type="checkbox"/> Producto 1 TOTAL: 1 <input type="checkbox"/> Producto 2 TOTAL: 1	SM Software, S.A. de C.V.	4
			Calcula

Figura IV.16 Vista Contabilización

Subsistema Asistente de Registro y Activación de Software

Como se determinó en el capítulo anterior, el objetivo de este subsistema es guiar al usuario final de una manera gráfica, paso a paso en el proceso de registro y activación del software de aplicación protegido contra la piratería.

Este subsistema debe ser llamado desde el programa de aplicación, por lo cual debe ser implantado antes de su compilación. La idea principal es que cada vez que inicie el programa de aplicación, éste llama al asistente para que verifique la existencia de la licencia de uso y poder continuar con la ejecución, de no existir se muestra un cuadro de diálogo que guía al usuario final en el proceso de obtención de la licencia de uso, si este cuadro de diálogo es cerrado o cancelado, por el usuario, se termina la ejecución del programa, es decir, se cierra por completo la aplicación. La licencia de uso consiste de una serie de códigos ocultos en el equipo de cómputo del usuario, por lo tanto, una vez generados dichos códigos, en lo sucesivo el asistente solo verifica su existencia. A continuación se muestra la secuencia de ejecución de un programa de aplicación con la integración del asistente.

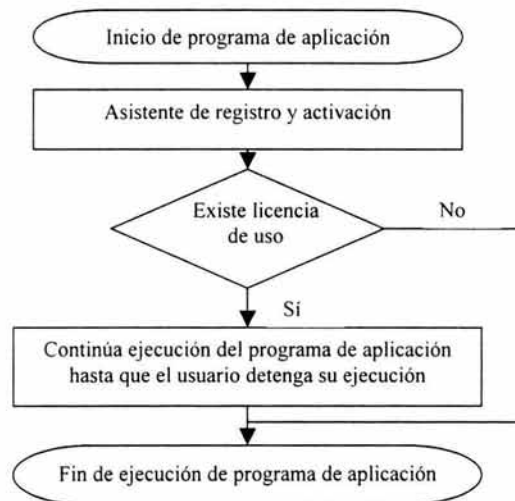


Figura IV.17 Integración del asistente de registro y activación de software

Como puede observarse en el diagrama anterior, el asistente determina la ejecución del programa de aplicación, verificando la existencia de la licencia de uso indica al mismo que continúe o detenga su ejecución. Dicho de otra forma el asistente funciona como interruptor que permita la activación del programa de aplicación. Además de ser el medio a través del cual se obtiene la licencia de uso.

Este subsistema está integrado como un componente de software creado en Visual Basic. Un componente de software creado con Visual Basic es un archivo que contiene código ejecutable (un archivo con extensión .exe, .dll u .ocx) y que proporciona objetos que pueden usar otras aplicaciones y componentes. Un objeto es una combinación de código y datos que se puede tratar como una unidad, es decir, es una entidad que incluye datos y el código que los maneja. Un objeto puede ser una parte de una aplicación, como un botón o un formulario, también puede ser un objeto una aplicación entera. El desarrollo de componentes de software mediante Visual Basic es conocido como la tecnología ActiveX. Esta tecnología no debe confundirse con la programación orientada a objetos (OOP⁷). La programación orientada a objetos es una forma de crear componentes de software basados en objetos; ActiveX es una tecnología que permite combinar componentes basados en objetos creados con muchas herramientas diferentes. Para decirlo de otro modo, la programación orientada a objetos se ocupa de crear objetos, mientras que ActiveX se ocupa de que los objetos funcionen juntos. Es importante mencionar que Visual Basic no es un lenguaje de programación orientado a objetos real, sino es un lenguaje basado en objetos.

Para crear el asistente de registro y activación de software, se utiliza un componente del tipo ActiveX DLL, este tipo puede servir a múltiples aplicaciones. La DLL (Dynamic Link Library) contiene al asistente considerado como un objeto con propiedades y métodos que pueden llamarse desde aplicaciones u otras DLL. Las propiedades se pueden considerar como sus atributos y los métodos como sus acciones. Los archivos DLL constituyen bibliotecas de procedimientos a los que las aplicaciones se pueden enlazar y usar en tiempo de ejecución, en lugar de enlazarse estáticamente en tiempo de compilación. Esto implica que las bibliotecas se pueden actualizar independientemente de la aplicación y muchas aplicaciones pueden compartir una única DLL. El propio entorno Microsoft Windows está formado por DLL, y otras aplicaciones llaman a los procedimientos incluidos en estas bibliotecas para mostrar ventanas y gráficos, administrar la memoria o realizar otras tareas. Estos procedimientos reciben el nombre de API de Windows o interfaz de programación de aplicaciones.

API es la abreviatura de **A**plication **P**rogramming **I**nterface. Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador, como por ejemplo, imprimir un carácter en pantalla, leer el teclado, escribir en un fichero de disco, etc. Visto desde la perspectiva del código máquina, el API aparece como una serie de llamadas, mientras que si lo vemos desde la de un lenguaje de alto nivel, el API aparece como un conjunto de procedimientos y funciones.

⁷ La programación orientada a objetos (OOP) es una forma de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación. En OOP Un programa se divide en componentes que contienen procedimientos y datos. Cada componente se considera un *objeto*. Un *objeto* es una *unidad que contiene datos y las funciones que operan sobre esos datos*.

El componente ActiveX DLL que se desarrollo para contener al asistente de registro y activación de software tiene las siguientes características:

Nombre de Archivo: AsistRA.dll

Nombre del objeto contenido: AsistRegistroActivacion

Propiedades: Las propiedades se pueden considerar como sus atributos, a continuación se describe cada una de sus propiedades

IDSoftware: Permite establecer el valor numérico que identifica el software a registrar y activar.

Sintaxis: `AsistRegistroActivacion.IDSoftware = <Número>`

El valor numérico debe ser del tipo entero comprendido en el intervalo de 0 a 999. Este valor corresponde directamente al `KeyTipoLic` devuelto por el "Subsistema generador y administrador de licencias de uso", para cada producto de software.

Ejemplo: Supóngase, que el producto de software "X" tiene asignado el número 5.

`AsistRegistroActivacion.IDSoftware = 5`

TituloLicencia: Permite establecer la cadena de texto que conforma el titulo de la licencia de uso a mostrar en el asistente.

Sintaxis: `AsistRegistroActivacion.TituloLicencia = [Cadena]`

La cadena de texto esta limitada a 110 caracteres. Este texto es mostrado en la parte superior del asistente, puede incluirse en el cualquier carácter, incluyendo caracteres especiales.

Ejemplo: El titulo para un producto de software "X", podría ser:

`AsistRegistroActivacion.TituloLicencia = "Licencia de uso para Software X "`

URLHttp: Permite establecer la cadena texto correspondiente a la dirección http del sitio en Internet donde el asistente localiza la base de datos de licencias de uso.

Sintaxis: `AsistRegistroActivacion.URLHttp = <Cadena>`

La cadena de texto esta limitada a 1000 caracteres. La dirección http corresponde con la misma dirección donde se encuentre la base de datos alojada en Internet, a través de esta dirección el asistente accesa a la base de

datos para el registro del usuario final y la obtención de la licencia de uso del software.

Ejemplo:

```
AsistRegistroActivacion.URLHttp="http://www.empresa_x.com.mx/usrs"
```

Métodos: A continuación se describe los métodos del asistente. Estos se pueden considerar como sus acciones

VerificaLicUso: Comprueba la licencia de uso para el programa de aplicación con identificador igual al definido en la propiedad "ID_Software", si no encontró la licencia de uso se muestra un cuadro de dialogo que guía al usuario final en el proceso de registro y activación para la obtención de la licencia de uso de software.

Sintaxis: `AsistRegistroActivacion.VerificaLicUso ([NoMostrarDialogo])`

El valor devuelto por este método es de tipo lógico, "Verdadero" para indicar que se ha comprobado satisfactoriamente la existencia de la licencia de uso, "Falso" denota lo contrario.

Parámetros

NoMostrarDialogo: Permite establecer de forma opcional si se muestra el cuadro dialogo del asistente. El valor asignado a esta propiedad debe ser de tipo lógico, "Verdadero" para indicar que se muestre el dialogo y "Falso" para que no se muestre. Útil para cuando sólo se desea hacer la verificación de la licencia de uso de forma transparente al usuario final.

Código de ejemplo de uso del asistente de registro: Debe ser agregado al programa de aplicación que se desea proteger y es llamado inmediatamente después de iniciar su ejecución.

```
Dim AsistRA As AsistRegistroActivacion
Set AsistRA = New AsistRegistroActivacion '//Creación del objeto
AsistRA.TituloLicencia = "Licencia de uso para software X"
AsistRA.NombreCia = "Cia. Desarrolladora de Software X"
AsistRA.UrlHttp = "http://www.empresa_x.com.mx/usuarios/"
AsistRA.IDSoftware = 2
If AsistRA.VerificaLicUso = False Then '//Comprobación
    End                               '// Fin de aplicación
End If
Set AsistRA = Nothing '// Eliminación del objeto
```

El código anterior verifica la existencia de licencia de uso para el software y de no encontrarse muestra el asistente de registro y activación para la obtención de la licencia de uso del software que lleva por título "X", desarrollado por la

compañía "X", la cual cuenta con su servidor de licencias de uso en la dirección de Internet "http://www.empresa_x.com.mx/usuarios/" y el identificador de software es "2". A continuación se describe paso a paso el proceso que sigue el asistente para la adquisición de la licencia de uso:

- *Bienvenida*

Pantalla inicial del asistente que da la bienvenida al usuario final indicando el nombre del producto a registrar y activar. Así mismo se muestra el nombre de la empresa que desarrollo el software. Contiene dos botones en la parte inferior que permiten cancelar el asistente o bien ir al siguiente paso que consiste en la solicitud del número de serie correspondiente al paquete del producto.

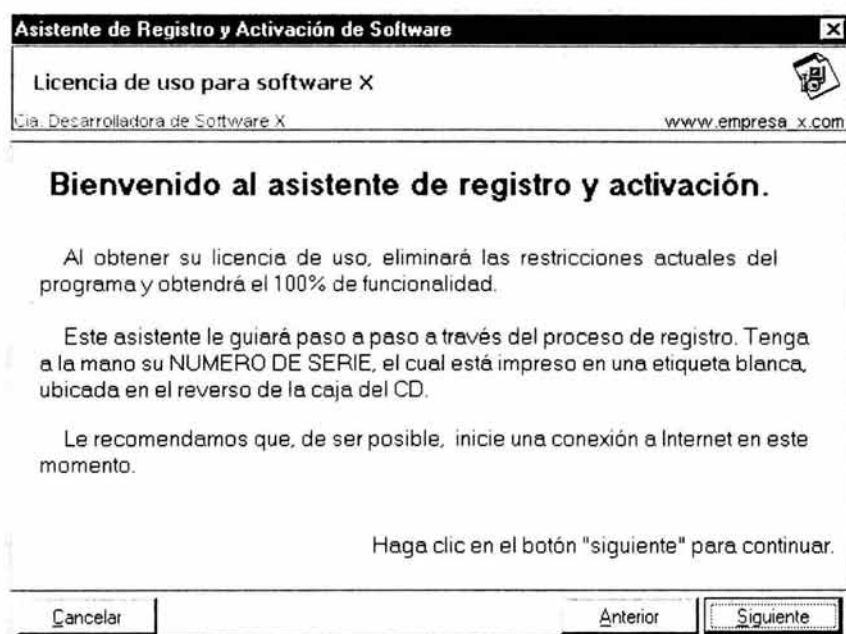


Figura IV.18 Primer paso del asistente de registro y activación de software.

- *Solicitud de número de serie.*

En este paso el usuario debe ingresar el número de serie proporcionado por el fabricante del software; para ello cuenta con cuatro cajas de texto separadas mediante un guión y con una longitud máxima de cinco caracteres en cada una de ellas, correspondiendo con el formato del número de serie proporcionado.



Figura IV.19 Solicitud de número de serie del paquete de software

Al ingresar caracteres que sean letras son convertidos de forma automática a su correspondiente letra mayúscula para su mejor visualización. Al completarse cada caja de texto el cursor se posiciona de forma automática en la siguiente caja de texto. Una vez ingresado el número de serie el usuario debe hacer clic en el botón siguiente para proceder con la validación del número de serie ingresado. Después de ser comprobada la validez del número de serie el asistente de registro y activación lleva al usuario al paso siguiente, que consiste en preguntar al usuario si posee conexión a Internet en el equipo, de lo contrario permanece en el mismo paso y se le indica al usuario, mediante un mensaje, que el número de serie ingresado no es válido.

- *Verificar conexión a Internet.*

En este paso se pregunta al usuario si dispone de una conexión a Internet en el equipo, pudiendo elegir entre las opciones de "Sí ó No". La siguiente figura muestra el ejemplo correspondiente.

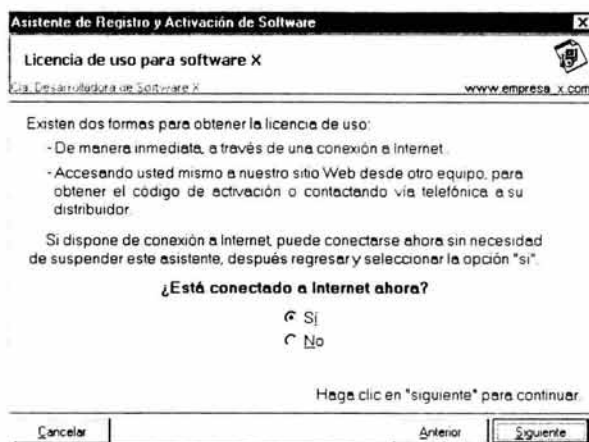


Figura IV.20 Comprobación de conexión a Internet

Este paso es crucial, debido a que si el usuario cuenta con conexión a Internet en su equipo, puede obtener de forma inmediata su licencia de uso, de lo contrario cuenta con dos formas alternativas, las cuales son:

- Acceder desde otro equipo al sitio Web de la empresa desarrolladora del software de aplicación, en la sección correspondiente a licencias de uso.
- Vía telefónica contactar a su distribuidor del programa, en horas y días hábiles, para que se le proporcione su licencia de uso.

A continuación se describe el proceso de obtención de licencia de uso para el caso en que el usuario final si cuenta con conexión a Internet; mas adelante se analiza el caso contrario.

- **El usuario si cuenta con conexión a Internet.**

Si el usuario dispone de conexión a Internet en el equipo debe iniciarla y una vez hecha la conexión debe hacer clic en el botón de siguiente para que el asistente proceda con la comprobación del número de serie a través del servidor de licencias de uso en Internet.

Comprobación de número de serie: El asistente envía el número de serie al servidor de licencias de uso en Internet y este se encarga de verificar en su base de datos la existencia del número de serie recibido devolviendo una respuesta de la existencia o inexistencia del mismo.

- Si no existe el número de serie en el servidor de licencias de uso en Internet, el asistente envía un mensaje al usuario, indicándole que el número de serie ingresado no existe y permanece en el mismo paso.

- Si se ha comprobado la existencia del número de serie se procede a verificar si el número de serie ya había sido registrado; si aún no ha sido registrado, el siguiente paso es el formulario de registro, como se muestra en la figura IV.21, después de que el usuario se registra, se le asigna un código de usuario y este se muestra en pantalla para que el usuario lo anote en la misma etiqueta que contiene el número de serie de su producto, la figura IV.22 muestra un ejemplo de un código de usuario asignado después de registrarse. Finalmente después de haberse registrado el usuario y se le haya asignado un código de usuario, el siguiente y último paso es un mensaje de agradecimiento, como lo muestra la figura IV.23

Figura IV.21 Formulario de registro de usuario final

Figura IV.22 Asignación de código de usuario

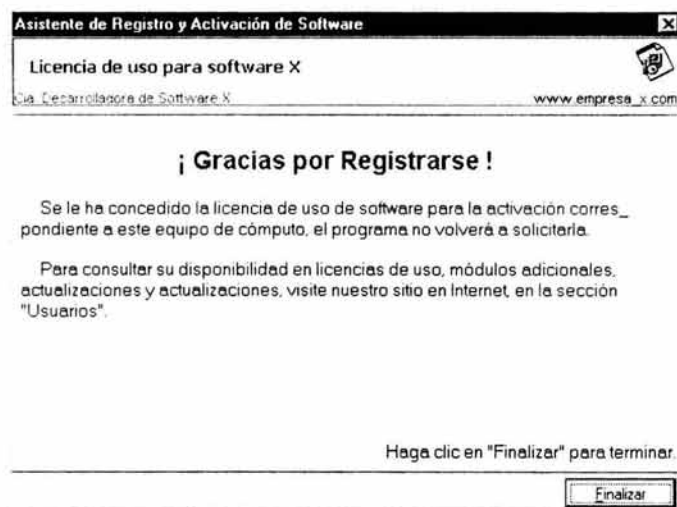


Figura IV.23 Agradecimiento

- Si se ha comprobado la existencia del número de serie proporcionado y dicho número de serie ya ha sido registrado el siguiente paso solicita el ingreso del código de usuario que se le otorgó cuando se registro, la figura IV.24 muestra un ejemplo. Si el código de usuario es correcto, el asistente procede a verificar si el usuario tiene la licencia de uso disponible en el servidor de licencias, si es el caso, el siguiente y último paso es un mensaje de agradecimiento, como lo muestra la figura IV.23, de lo contrario se muestra un mensaje indicando que no cuenta con licencia de uso de software disponible.

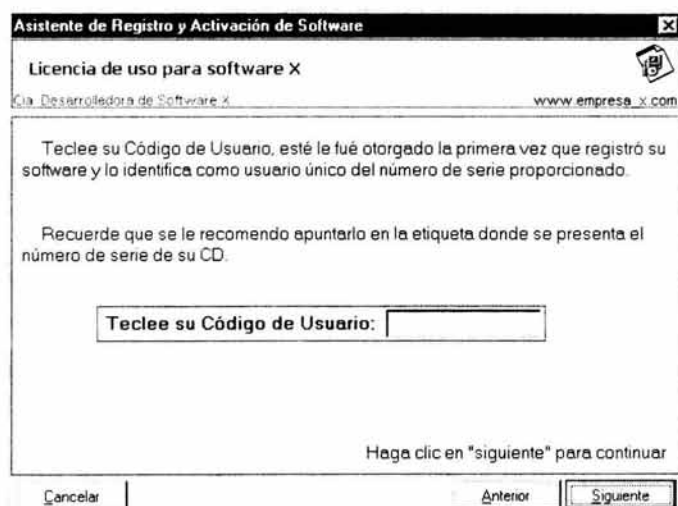


Figura IV.24 Ingresar código de usuario

Ahora se analiza el caso en el que el usuario no cuenta con conexión a Internet.

- *El usuario no cuenta con conexión a Internet.*

Si el usuario no dispone de conexión a Internet en el equipo debe, seleccionar la opción "No" (paso mostrado en la figura IV.25) y hacer clic en el botón de siguiente. El asistente procede a generar un código de equipo, y solicita se ingrese una clave de acceso. La figura siguiente muestra un ejemplo de este paso.

Figura IV.25 Solicitud clave de acceso

Como se indica en el mensaje de la figura anterior, el usuario puede obtener la clave de acceso solicitada, visitando el sitio Web de la empresa que desarrollo el programa de aplicación y acceder a la sección de usuarios; o bien puede contactar vía telefónica a su distribuidor del programa de aplicación. Ya sea por el sitio Web o a través del distribuidor, el usuario debe proporcionar su número de serie, su código de usuario (en el caso que ya hubiese se registrado con anterioridad) y el código de equipo generado por el asistente.

Habiendo obtenido el usuario, la clave de acceso debe ingresarla y si esta es válida se obtiene la licencia de uso y con ello obtiene la activación de su software. La figura IV.23 muestra el paso final después de haber validado con éxito la clave de acceso.

Hasta este punto se ha diseñado el asistente de registro y activación de software, se ha presentado a través de un ejemplo su forma de integración y se ha definido a detalle la vista de usuario de todo el proceso de registro del software de aplicación y la obtención de la correspondiente licencia de uso para la activación del mismo.

Ahora se desarrollan a detalle los procedimientos y funciones que el asistente emplea para:

- Validación de número de serie
- Comprobación de existencia de número de serie
- Comunicación con el servidor de licencias de uso
- Registro de usuario
- Obtención de código de usuario
- Validación de código de usuario
- Obtención de licencia de uso
- Activación de software
- Generación de código de equipo

Cada uno de estos puntos es descrito a continuación.

Validación de número de serie

Como se ha descrito en el apartado correspondiente al “Generador y administrador de licencias de uso” el número de serie esta conformado por 20 caracteres alfanuméricos, de ellos, 10 caracteres contienen codificado un número de 10 dígitos, único para cada producto, y los restantes 10 caracteres son un código de verificación. Debe recordarse que los caracteres del número de serie fueron generados a través de cálculos matemáticos entre los dígitos del número único y codificado a partir de una tabla de asignación que designa una letra del alfabeto a los valores numéricos del 10 al 35. Por consiguiente el proceso de validación consiste en decodificar los caracteres alfabéticos, a su correspondiente valor numérico, empleando la misma tabla de asignación y realizar los cálculos matemáticos inversos para comprobar la autenticidad del número de serie ingresado por el usuario.

A continuación se describe el algoritmo para comprobar la validez del número de serie:

- Recuperar el número de serie ingresado por el usuario.
- Comprobar su longitud de 20 caracteres. Una longitud menor lo invalida.
- Decodificar caracteres alfabéticos su correspondiente valor numérico según tabla de asignación utilizada para su creación.
- Recuperar los 10 dígitos que corresponden al número entero aleatorio y realizar cálculos inversos para decodificarlo. Al primer resultado incorrecto se determina la invalidez del número de serie.

- Recuperar los 10 dígitos restantes, los cuales constituyen un código de verificación, y realizar cálculos inversos para decodificarlo. Al primer resultado incorrecto se determina la invalidez del número de serie.

Si se ha comprobado exitosamente la validez del número de serie ingresado por el usuario, debe comprobarse que dicho número ya ha sido distribuido por la empresa desarrolladora del programa de aplicación a registrar. El proceso de comprobación es descrito a continuación.

Comprobación de existencia de número de serie

Esta comprobación del número de serie consiste en verificar, en el servidor de licencias de uso ubicado en Internet, si existe el número ingresado por el usuario. Como se definió en el diseño del subsistema "Generador y administrador de licencias de uso", cada número de serie generado debe ser agregado en la base de datos alojada en Internet, así como también su licencia de uso correspondiente.

Para la comprobación del número de serie, desde el asistente, es indispensable que exista una conexión a Internet, ya que este es el medio de comunicación que el asistente emplea para realizar la petición de comprobación al "servidor de licencias de uso en Internet". La petición es hecha a través de una página Web, la cual forma parte del servidor de licencias y es quien se encarga de realizar la consulta en la base de datos y redireccionar hacia otra página como resultado de la comprobación. La comunicación del asistente con el servidor de licencias de uso en Internet se expone a continuación.

Comunicación con el servidor de licencias de uso

Una conexión a Internet es el medio de comunicación que el asistente requiere para la realización de peticiones al servidor de licencias de uso. La petición es a través de páginas Web específicas, dichas páginas son del tipo ASP y tienen funciones bien definidas para recibir la petición que se les encomiende, disponen de la capacidad de realizar inserción, actualización, eliminación y consulta de registros de la base de datos alojada en el mismo sitio de Internet donde se encuentran.

El asistente contempla en su código los nombres de las páginas para cada petición, así mismo los parámetros solicitados. La forma de respuesta de la página al asistente es a través de un redireccionamiento hacia una nueva página que incluye en dirección información devuelta al asistente. En el apartado correspondiente al "Servidor de licencias de uso en Internet", abordado más adelante, se expone a detalle su funcionamiento.

Registro de usuario

Después de comprobar la existencia del número de serie en el servidor de licencias de uso en Internet, el asistente solicita se verifique si el número ya fue registrado, de no estar registrado aún, se procede con la realización de registro de usuario para el mismo número de serie.

El registro del usuario desde el asistente constituye una petición hacia el servidor de licencias de uso en Internet, básicamente consiste de una página Web conteniendo un formulario de registro, solicitando datos personales del usuario final que registra por primera y única vez el número de serie correspondiente al paquete de software. La validación de la información ingresada en el formulario es validada por la misma página, así mismo se encarga de enviar la información hacia la base de datos alojada en Internet, insertándose en ésta un nuevo registro de datos y asignándose automáticamente un número entero aleatorio único.

El número entero aleatorio esta dentro del intervalo de -2147483648 a 2147483648 y constituye la llave única de identificación del usuario del número de serie ingresado y este número debe ser asignado al número de serie dentro de la misma base de datos en la tabla de datos correspondiente. Además es considerado como el "*Código de Usuario*" asignado, es único, es decir, no hay duplicidad y esta relacionado únicamente con sólo un número de serie.

Código de usuario

El código de usuario es el identificador del dueño único de un número de serie registrado. Consiste de un número entero en el intervalo de -2147483648 a 2147483648, para valores negativos se le antecede una letra "A" mayúscula, en representación del signo negativo, por ejemplo un código de usuario podría ser: "A312564", que corresponde al número -312564. El número que representa el código de usuario, es generado de forma aleatoria por el servidor de licencias de uso en Internet, justo en el momento que el usuario del software de aplicación registra su copia de software; como ya se ha comentado es único, y esta relacionado únicamente con solo un número de serie. Cada número de serie tiene solamente un dueño, lo cual quiere decir que puede ser registrado sólo una vez.

Obtención de código de usuario

La obtención del código de usuario es únicamente a través del registro del usuario mediante un formulario, donde se le solicitan datos personales y de dirección. Este formulario esta en una página Web, como se ha descrito con anterioridad, al enviar el formulario se envía la información al servidor de licencias de uso en Internet y si la información fue recibida satisfactoriamente junto con el número de serie, se muestra en una nueva página el código de usuario asignado. De esta forma el asistente obtiene y muestra al usuario su código asignado.

Validación de código de usuario

Después de comprobar la existencia del número de serie en el servidor de licencias de uso en Internet, el asistente solicita se verifique si el número ya fue registrado, de ser así, el asistente solicita se ingrese el código de usuario que se le fue asignado cuando registro su copia de software.

La validación del código de usuario es a través de una petición del asistente hacia el servidor de licencias de uso en Internet, en la petición se envía el código de usuario y el número de serie. El servidor se encarga de verificar si existe el código de usuario y si corresponde con el número de serie.

Obtención de licencia de uso

Si el número de serie no había sido registrado, es de suponerse que cuenta con licencia de uso disponible, por consiguiente después de que el usuario recibe su código de usuario, el asistente realiza la petición de la licencia de uso correspondiente.

En el caso de que el número de serie ya fue registrado, el asistente solicita se ingrese el código de usuario, después de su validación, el asistente solicita al servidor verificar si hay licencia de uso disponible, de existir se realiza la petición de la misma.

La obtención de la licencia de uso, consiste esencialmente en actualizar un registro de la base de datos del servidor de licencias en Internet, el registro de datos es actualizado con la fecha en que fue concedida y ello significa que ya no puede volverse a conceder, después de actualizar el registro se responde al asistente que se ha concedido exitosamente la licencia de uso. Finalmente después de haberse obtenido la licencia de uso, el asistente procede con la activación del programa de aplicación.

Activación de software

Este proceso realiza la generación de códigos ocultos en el equipo de cómputo donde se ha concedido la licencia de uso del software de aplicación, podría decirse que en cierta forma este conjunto de códigos constituye la licencia de uso obtenida, ya que el asistente verifica la existencia de los mismos para poder permitir la ejecución completa del programa de aplicación. A continuación se describe el algoritmo que activa el software.

Algoritmo de activación de software

- Generación de archivo oculto dentro de la carpeta del sistema operativo
- Abrir archivo generado para acceso binario ⁷
- Inserción en el archivo de 250 Kb de datos aleatorios (información sin un sentido lógico)
- Recuperación de número de serie de disco duro
 - Inserción en el archivo de número de serie de disco duro codificado, distribuyendo cada dígito en diferentes puntos del archivo.
- Generar un código con la fecha de creación de archivo
 - Inserción en el archivo del código generado, a partir de un byte específico
- Generación aleatoria de una clave de registro dentro de una clave específica de Windows
 - Inserción en el archivo de clave de registro aleatoria, a partir de un byte específico
- Generación aleatoria de una segunda clave de registro dentro de una clave específica de Windows
- Generación de valor numérico aleatorio
 - Inserción de valor numérico en segunda clave de registro
 - Inserción en el archivo de valor numérico, a partir de un byte específico
- Generar un código con la fecha de modificación de archivo
 - Inserción en el archivo del código generado, a partir de un byte específico
- Cerrar archivo

Se ha presentado el algoritmo para la activación del software, por lo tanto el asistente de activación realiza una verificación del archivo y claves de registro de Windows cada vez que se inicie el programa de aplicación, si la verificación es satisfactoria permite la ejecución completa del programa de aplicación, de lo contrario solicita la licencia de uso correspondiente.

⁷ **Acceso Binario:** Forma de abrir un archivo sin tomar en cuenta la estructura que tiene éste ni lo que significan cada uno de sus bytes. El archivo se trata como una secuencia de bytes

Dentro del algoritmo se hace mención del registro de Windows, a continuación se da una descripción para su mayor entendimiento, en cuanto a la manipulación de las claves que almacena.

Registro de Windows

Definición: Es una base de datos con una estructura jerárquica donde las aplicaciones pueden guardar información sobre su configuración y el sistema operativo almacena información sobre si mismo.

Antes, las aplicaciones y el sistema operativo utilizaba los archivos “.INI”, pero su proliferación y su escasa estructuración hicieron que Microsoft se plantease dar un sistema mucho más práctico para guardar los datos de configuración de las aplicaciones.

El primer Registro se incluye en Windows 3.1, pero es en Windows NT y Windows 95 cuando se nos ofrece un auténtico registro de configuraciones.

Estructura del Registro

El Registro tiene una estructura jerárquica. Está jerarquía es basada en una serie de claves (keys) que pueden contener valores u otras claves.

Las claves pueden contener tantos valores como se quiera, identificados cada uno de ellos por una etiqueta.

Si se quiere ver el registro de configuraciones debe ejecutar debe ejecutar REGEDIT para el caso de Windows NT y 2000 debe ejecutar REGEDIT32.

Claves principales

En Windows 95 y Windows NT el Registro tiene una conjunto de entradas principales colgando de una raíz, denominada ROOT. La lista de las entradas de Windows 95 y Windows NT se muestran a continuación.

Win16 y Win32s

HKEY_CLASSES_ROOT - ficheros registrados y OLE

Win95,98,XP

HKEY_CLASSES_ROOT - por compatibilidad con Win16

HKEY_CURRENT_USER - configuración del usuario actual

HKEY_LOCAL_MACHINE - configuración de esta máquina

HKEY_USERS - configuración por defecto de usuarios

HKEY_CURRENT_CONFIG - configuración actual de esta máquina
 HKEY_DYN_DATA - estado dinámico de esta máquina

WinNT

HKEY_CLASSES_ROOT - por compatibilidad con Win16
 HKEY_CURRENT_USER - configuración del usuario actual
 HKEY_LOCAL_MACHINE - configuración de esta máquina
 HKEY_USERS - configuración por defecto de usuarios

Existen dos entradas en el Registro que sólo están disponibles en Windows 95: HKEY_CURRENT_CONFIG y HKEY_DYN_DATA. Estas entradas están especialmente pensadas para obtener información sobre dispositivos Plug&Play y para gestionar la conexión y desconexión de dispositivos en portátiles.

No se va a describir aquí la información que podemos encontrar en las distintas entradas del Registro, pues nos centraremos en las funciones de acceso al Registro. Para obtener información sobre la información a la que podemos acceder lo mejor es consultar los "Resource Kit" de Windows.

Tipos de información

Todos los datos que pueden almacenarse en las distintas entradas de este Registro de Windows son de tipo carácter y, como hemos dicho, sólo puede haber uno por clave.

La información almacenada en el Registro pueden ser de los siguientes tipos:

Tipo no definido Constante REG_NONE = 0	Enlace simbólico de Unicode Constante REG_LINK = 6
Cadena de caracteres Constante REG_SZ = 1	Matriz de cadenas Constante REG_MULTI_SZ = 7
Binario Constante REG_BINARY = 3	Lista de recursos de dispositivo Constante REG_RESOURCE_LIST = 8
Número de 32 bits Constante REG_DWORD = 4	

Continuemos con el último procedimiento que el asistente de registro emplea para lograr la activación del software cuando el usuario final no dispone de una conexión a Internet en su equipo de cómputo.

Las siguientes funciones escritas en Visual Basic permiten la manipulación del registro de Windows mediante llamadas a funciones del API de Windows:

```
Public Const HKEY_CLASSES_ROOT = &H80000000
Public Const HKEY_LOCAL_MACHINE = &H80000002

Public Const REG_SZ = 1 ' Cadena Unicode terminada en null
Public Const REG_BINARY = 3 ' Free form binary
Public Const REG_DWORD = 4 ' Número de 32 bits

Private Declare Function RegCloseKey Lib "advapi32.dll" (ByVal hKey As Long) As Long
Private Declare Function RegCreateKey Lib "advapi32.dll" Alias "RegCreateKeyA" (ByVal hKey
As Long, ByVal lpSubKey As String, phkResult As Long) As Long
Private Declare Function RegDeleteValue Lib "advapi32.dll" Alias "RegDeleteValueA" (ByVal
hKey As Long, ByVal lpValueName As String) As Long
Private Declare Function RegDeleteKey Lib "advapi32.dll" Alias "RegDeleteKeyA" (ByVal hKey
As Long, ByVal lpSubKey As String) As Long
Private Declare Function RegOpenKey Lib "advapi32.dll" Alias "RegOpenKeyA" (ByVal hKey As
Long, ByVal lpSubKey As String, phkResult As Long) As Long
Private Declare Function RegOpenKeyEx Lib "advapi32.dll" Alias "RegOpenKeyExA" (ByVal
hKey As Long, ByVal lpSubKey As String, ByVal Reserved As Long, ByVal samDesired As Long,
phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32.dll" Alias "RegQueryValueExA"
(ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, lpType As
Long, lpData As Any, lpcbData As Long) As Long
Private Declare Function RegSetValueEx Lib "advapi32.dll" Alias "RegSetValueExA" (ByVal hKey
As Long, ByVal lpValueName As String, ByVal Reserved As Long, ByVal dwType As Long,
lpData As Any, ByVal cbData As Long) As Long

Const STANDARD_RIGHTS_ALL = &H1F0000
Const SYNCHRONIZE = &H100000
Const READ_CONTROL = &H20000
Const KEY_CREATE_LINK = &H20
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_ALL_ACCESS = ((STANDARD_RIGHTS_ALL Or KEY_QUERY_VALUE Or
KEY_SET_VALUE Or KEY_CREATE_SUB_KEY Or KEY_ENUMERATE_SUB_KEYS Or
KEY_NOTIFY Or KEY_CREATE_LINK) And (Not SYNCHRONIZE))
'=====
' Procedimiento: SaveStringData
' Guarda un nuevo valor de tipo alfanumérico en el registro de windows
'=====
Sub SaveStringData(hKey As Long, strPath As String, strValue As String, strData As String)
    Dim ret
    RegCreateKey hKey, strPath, ret
    RegSetValueEx ret, strValue, 0, REG_SZ, ByVal strData, Len(strData)
    RegCloseKey ret
End Sub
```

```

=====
' Procedimiento: SaveStringLong
' Guarda un nuevo valor de tipo entero largo en el registro de windows
=====
Sub SaveStringLong(hKey As Long, strPath As String, strValue As String, strData As String)
    Dim ret
    RegCreateKey hKey, strPath, ret
    RegSetValueEx ret, strValue, 0, REG_BINARY, CByte(strData), 4
    RegCloseKey ret
End Sub

=====

' Procedimiento: ExistKey
' Verifica si existe una clave de registro
=====
Function ExistKey(hKey As Long, NameKey As String) As Boolean
    Dim ret As Long
    RegOpenKeyEx hKey, NameKey, 0, KEY_ALL_ACCESS, ret
    If ret <> 0 Then ExistKey = True
    RegCloseKey ret
End Function

=====

' Funcion: GetString
' Obtiene un valor del registro de windows
=====
Public Function GetString(hKey As Long, strPath As String, strValue As String)
    Dim ret
    RegOpenKey hKey, strPath, ret
    GetString = RegQueryStringValue(ret, strValue)
    RegCloseKey ret
End Function

=====

' Funcion: DelSetting
' Elimina un valor del registro de windows
=====
Public Sub DelSetting(hKey As Long, strPath As String, strValue As String)
    Dim ret
    RegCreateKey hKey, strPath, ret
    RegDeleteValue ret, strValue
    RegCloseKey ret
End Sub

=====

' Funcion: RegQueryStringValue
' Consulta un valor del registro de windows
=====
Private Function RegQueryStringValue(ByVal hKey As Long, ByVal strValueName As String) As String
    Dim lResult As Long, lValueType As Long, strBuf As String, lDataBufSize As Long

```

```
IResult = RegQueryValueEx(hKey, strValueName, 0, IValueType, ByVal 0, IDataBufSize)
If IResult = 0 Then
  If IValueType = REG_SZ Then
    strBuf = String(IDataBufSize, Chr$(0))
    IResult = RegQueryValueEx(hKey, strValueName, 0, 0, ByVal strBuf, IDataBufSize)
    If IResult = 0 Then
      RegQueryStringValue = Left$(strBuf, InStr(1, strBuf, Chr$(0)) - 1)
    End If
  ElseIf IValueType = REG_BINARY Then
    Dim strData As Integer
    IResult = RegQueryValueEx(hKey, strValueName, 0, 0, strData, IDataBufSize)
    If IResult = 0 Then
      RegQueryStringValue = strData
    End If
  End If
End If
End If
End Function
```

Generación de código de equipo

El objetivo principal del código de equipo es identificar de manera única el equipo de cómputo donde se ha instalado software de aplicación y para la generación de un código, al que llamaremos “Clave de Acceso”, que permite la activación del software, pero esta clave de acceso es únicamente válida para el equipo de cómputo del que se obtuvo el código de equipo. Este es generado mediante un algoritmo que realiza la codificación de algunos dígitos del número de serie del disco duro sobre el que se encuentra el sistema operativo, el identificador de software y coloca elementos que permitan verificar su validez, es decir, que el mínimo cambio de uno de estos elementos lo invalide.

El código de equipo esta conformado por dos series de seis caracteres alfanuméricos, como se muestra a continuación: **XXXXXX - XXXXXX**

Donde la “X” puede ser cualquier dígito del cero al nueve o una letra del alfabeto, es decir, cualquier letra entre la “A” y la “Z”, exceptuando la letra “Ñ”. Considerando que cada carácter del número de serie puede tomar 35 diferentes valores y son 12 caracteres los que conforman al número de serie, por lo tanto se tienen 35^{12} posibles combinaciones distintas.

Un ejemplo real podría ser: **ZTDH8A - N9R2ZH**

A continuación se presenta el algoritmo para la generación del código de equipo, es esencial que la codificación del código emplee la misma tabla de asignación utilizada en la codificación del número de serie.

Algoritmo para generación de código de equipo

- Obtener el número de serie del disco duro sobre el que esta el sistema operativo
- Generar 3 valores numéricos, mediante operaciones matemáticas entre los dígitos del número de serie del disco duro, pero con la restricción de que el resultado de la operación sea un número entero comprendido dentro del intervalo de 0 a 35.
- Generar 3 valores numéricos, mediante operaciones matemáticas entre los dígitos del ID de software, pero con la restricción de que el resultado de la operación sea un número entero comprendido dentro del intervalo de 0 a 35.
- Generar 6 valores numéricos donde cada número del arreglo es el resultado de operaciones matemáticas entre los 6 valores numéricos antes generados y de igual forma con la restricción de que el resultado sea un número entero comprendido dentro del intervalo de 0 a 35. Básicamente estos 6 valores adicionales son el código de verificación de autenticidad.
- Codificar cada uno de los 12 valores numéricos generados, a su correspondiente dígito o letra según sea su valor y de acuerdo a la misma tabla de asignación utilizada en la generación del número de serie del software de aplicación.

Servidor de licencias de uso de software en Internet

Este Servidor esta constituido principalmente por páginas Web ASP alojadas en un sitio en Internet, esencialmente contiene los procedimientos y funciones que atienden y dan respuesta a las peticiones solicitadas; estas páginas se encuentran almacenadas dentro de un directorio nombrado "actsoft" en el cual se aloja la base de datos BD_LICS.mdb, en ella se almacena la información generada por el subsistema "Generador y administrador de licencias de uso de software", "Subsistema Asistente de activación de software" y por el "Subsistema de Registro, obtención y consulta de licencias de uso en Internet".

Como se estableció en la etapa de análisis, esta base de datos mantiene información correspondiente a licencias de uso, paquetes, licencias adicionales, licencias por reposición y datos de registro del usuario final del software.

Es importante mencionar que el sitio en Internet debe estar alojado en un "Servidor seguro". Para comprender mejor el concepto de lo que es un servidor seguro a continuación se presenta una definición.

Un servidor seguro es un servidor de páginas html, especialmente configurado para establecer una conexión transparente con el cliente, consiguiendo que la información que circule entre ellos (cliente-servidor) viaje a través de Internet encriptada mediante algoritmos que aseguran que sea inteligible sólo para el servidor y el visualizador que accede al web.

El término cliente/servidor describe un sistema en el que una máquina cliente solicita a una segunda máquina llamada servidor que ejecute una tarea específica. El cliente suele ser una computadora personal común conectada a una LAN⁸, y el servidor es, por lo general, una máquina anfitriona.

El programa cliente cumple dos funciones distintas:

- Gestiona la comunicación con el servidor.
- Solicita un servicio y recibe los datos enviados.

Por otro lado, maneja la interfaz con el usuario: presenta los datos en el formato adecuado y brinda las herramientas y comandos necesarios para que el usuario pueda utilizar las prestaciones del servidor de forma sencilla.

⁸ LAN: Son redes de propiedad privada, de hasta unos cuantos kilómetros de extensión. Por ejemplo una oficina o un centro educativo. Se usan para conectar computadoras personales o estaciones de trabajo, con objeto de compartir recursos e intercambiar información. Están restringidas en tamaño, lo cual significa que el tiempo de transmisión, en el peor de los casos, se conoce, lo que permite cierto tipo de diseños (deterministas) que de otro modo podrían resultar ineficientes. Además, simplifica la administración de la red. Suelen emplear tecnología de difusión mediante un cable sencillo al que están conectadas todas las máquinas.

El programa servidor en cambio, básicamente sólo tiene que encargarse de transmitir la información de forma eficiente. No tiene que atender al usuario. De esta forma un mismo servidor puede atender a varios clientes al mismo tiempo.

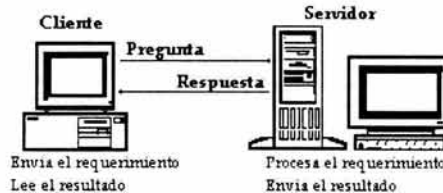


Figura IV.26 Cliente-Servidor

El navegador Netscape o Explorer, colaborando con el servidor seguro al que llama, encripta los datos de forma que, si algún individuo en el proceso de transmisión consigue apropiarse de éstos, no podrá leerlos ya que no dispone de la clave necesaria. Esta encriptación se basa en el protocolo SSL estándar desarrollado por Netscape Communications para transferir información segura a través de internet.

El protocolo **SSL (Secure Socket Layer)** es un protocolo de intercambio de información que permite asegurar la autenticación, confidencialidad e integridad de los datos que se transmiten a través de Internet. Para que la información alojada en un dominio pueda verse protegida bajo el protocolo SSL es necesario instalar un Certificado de seguridad en dicho dominio

Un Certificado de seguridad es un conjunto de documentos electrónicos emitidos por una entidad certificadora, que permiten encriptar la información transmitida e identificar a la fuente de dicha información. Para que el certificado sea fiable, la citada entidad debe ser un organismo de confianza capaz de garantizar la procedencia de la información, que es lo que en cierta medida proporciona la seguridad. Dos de las entidades certificadoras más conocidas son **Thawte** y **Verisign**. Un servidor seguro certificado por Verisign cuenta con una clave de 128 bits, con una parte secreta de 40 bits. Esto quiere decir que si algún intruso intenta descifrar los datos transmitidos con este sistema, debe realizar 240 complicadas operaciones para descifrar estos datos, que en tiempo de computación supone miles de años en una de las máquinas más potentes del mercado.


Cuando el usuario llena un formulario y pulsa el botón enviar, está enviando esos datos a través de la red. Estos datos, son transmitidos de servidor en servidor hasta llegar a su destinatario. Podrían ser recogidos (robados) en cualquiera de los servidores por los cuales pasan hasta llegar a su destino. Un servidor seguro garantiza la privacidad de datos que se transmiten por la red.

Por otro lado teniendo en cuenta la enorme cantidad de información que circula por el Internet, es muy poco probable, que la información transmitida a través de un web site a un usuario o viceversa sea interceptada por personas ajenas a la transacción. La posibilidad de una intercepción con fines dolosos es muchísimo menor, o prácticamente nula, cuando se utiliza un servidor seguro. El incremento espectacular de las ventas por Internet en todo el mundo no sería posible si las transacciones no fueran tan seguras (o más) que las realizadas por cualquier medio convencional.


El usuario puede saber si se encuentra en un servidor seguro sí:

- La dirección URL comienza por `https://` en vez de `http://`

En **Netscape** se presentan las siguientes indicaciones:

- La llave de la parte inferior izquierda, que habitualmente aparece partida se ve completa 
- Aparece una línea azul en el límite superior del área de visualización.

En **Microsoft Internet Explorer** :

- Aparece un candado cerrado como este 

Finalmente un **sitio seguro** es aquel que se instala sobre un servidor que dispone de los programas necesarios para implementar el protocolo de seguridad, que permite que toda la información viaje encriptada desde el cliente al servidor o viceversa. Los servidores y clientes de Internet utilizan un sistema de criptografía basado en claves públicas para obtener un sistema seguro de envío y recepción de información. Solo el servidor y el navegador pueden encriptar y desencriptar su información. Durante el tránsito por Internet, la información es un conjunto de datos sin sentido e indescifrable. Un tercero puede hipotéticamente llegar a ver los datos, pero no podrá interpretarlos. La información permanece segura y privada.

Por otro lado en dicho sitio web se encuentran alojadas las páginas “ASP” que sirven de intermediarias para realizar el intercambio de información entre los subsistemas y la base de datos aquí alojada, es decir, a través de dichas páginas se puede realizar altas, bajas, cambios y consultas de la información.

Debido a esto, la seguridad en el proceso de enviar y recibir información de y hacia el servidor es de gran importancia. Por ello es importante mencionar que ASP tiene muchas ventajas. La mayor es que reemplaza la forma tradicional de intercambiar información entre usuarios. Los métodos tradicionales de intercambio cliente-servidor involucraban la tecnología Common Gateway Interface (CGI). CGI requiere una aplicación separada desde un documento web para procesar todos los

datos regresados al servidor. Esto tiene desventajas tanto en la seguridad como en el procesamiento de recursos.

Active Server Pages permiten que el código sea incrustado en un documento HTML y que corra en el servidor. El código que necesita correr en el servidor se coloca entre especiales script tags. Estas tags le dicen al servidor que no envíe el código que está entre ellas al cliente. El código no es necesario hasta que se reciba una solicitud del cliente, luego éste procesa el código en el servidor basado en la información enviada de regreso desde el cliente, el código nunca deja el servidor haciendo por lo tanto a ASP muy seguro y fácil de programar. ASP se ejecuta sobre el servidor y envía datos de regreso al cliente (a través del browser), es compatible con múltiples plataformas. Existen varios lenguajes que se pueden usar para hacer ASP. El más comúnmente utilizado es VBScript y es por mucho el más fácil, ya que es nativo de Microsoft.

Para comprender mejor el flujo de información que existe entre los subsistemas y el servidor a continuación se presenta la siguiente figura.

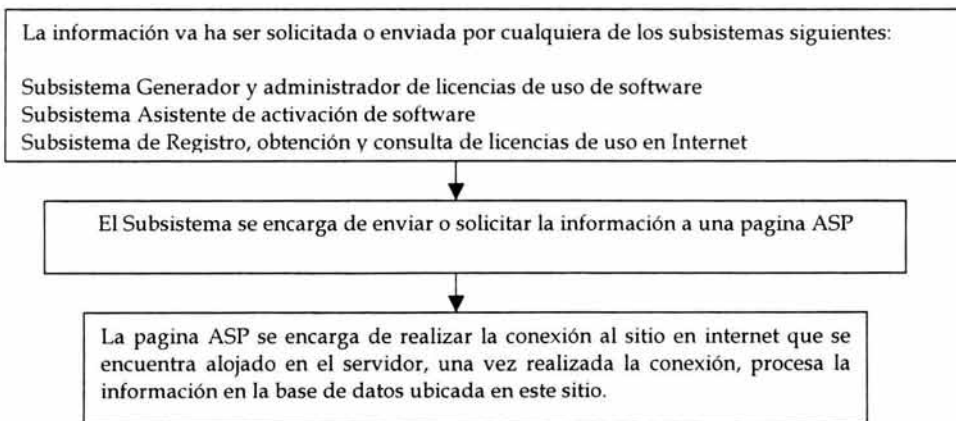


Figura IV.27 Flujo de Información

Para que una página web pueda desplegar dinámicamente los resultados de una Base de datos, necesita una conexión activa. Esta conexión es establecida por medio de un DSN.

Los controladores de bases de datos (programas que pasan información desde la aplicación Web a una base de datos) utilizan un Nombre de origen de datos - Data Source Name (DSN) para encontrar e identificar una base de datos ODBC en particular. Normalmente, el DSN contiene información de configuración de la base de datos, seguridad de usuarios y ubicación física, y puede tener la

forma de una entrada en el registro del sistema de Windows NT o de un archivo de texto.

Con ODBC puede elegir el tipo de DSN que va a crear: *Usuario, Sistema o Archivo*.

Los DSN de Usuario y de Sistema residen en el registro del sistema de Windows. Los DSN de Sistema permiten que todos los usuarios que han iniciado una sesión en un servidor concreto tengan acceso a una base de datos, mientras que los DSN de Usuario limitan la conectividad con la base de datos a los usuarios que dispongan de las credenciales de seguridad apropiadas. Los DSN de Archivo, que tienen la forma de archivos de texto, proporcionan acceso a varios usuarios y son fácilmente transferibles entre un servidor y otro mediante la copia de los archivos DSN.

Una vez creado el DSN, se debe indicar en la página ASP cómo se puede acceder a la Base de Datos. Cuando se crea una página, la ejecución del código tiene lugar en el servidor. Para diferenciar lo que es código en el lado del servidor de lo que es código en el lado del cliente, se utilizan los delimitadores `<%` y `%>`.

En este caso el DSN tiene el nombre "dsn_bd_actsoft" el cual es creado en el servidor.

En la carpeta "actsoft" del sitio en internet de este proyecto, se aloja la página "CnxDataBase.asp", la cual realiza la conexión a la base de datos y asigna la información del DSN a una variable de tipo cadena, como se muestra a continuación:

```
<% MM_CnxDataBase_STRING = "dsn=dsn_bd_actsoft;" %>
```

De esta forma cuando se requiera abrir una conexión a la base de datos con un objeto recordset, solo debe asignar esta variable a la propiedad `ActiveConnection`.

Para mayor claridad del proceso de petición de datos entre el subsistema y la base de datos alojada en el sitio de internet, a continuación, se muestra un ejemplo:

En el subsistema generador y administrador de licencias de uso de software, se va a realizar una salida de la empresa del producto con número de inventario 2, se da de alta este nuevo número de inventario en la base de datos alojada en el sitio en internet para que el usuario final pueda registrar el producto para su uso.

Primeramente la información es procesada en la base de datos local (Ver Módulo de Salidas de paquetes de software) y después enviada a través de la página "gl_salidapaquete.asp" a la base de datos a internet, de acuerdo a esto, el envío de información quedaría de la siguiente manera:

```
"gl_salidapaquete.asp? CodCD=" & CodCD & "& NumSerie=" & NumSerie &
"&Fecha=" & Format(Fecha, "mm/dd/yyyy") & "&ID_Paquete=" & ID_Paquete
```

Las palabras en negritas son las variables que reciben la información en la página ASP.

Antes de continuar con el proceso que sigue esta página al recibir la información a procesar, se hace mención teóricamente de su estructura, ya que es importante para una mejor comprensión de su código interno.

Primeramente, se debe tomar en cuenta que, para establecer la conexión con el origen de datos, ADO proporciona el objeto **Connection**, que puede utilizar para establecer y administrar las conexiones entre las aplicaciones y las bases de datos de ODBC. El objeto Connection incorpora diversas propiedades y métodos que puede utilizar para abrir y cerrar conexiones con bases de datos, y para enviar consultas de actualización de la información.

Como con el objeto Connection no se pueden crear secuencias de comandos que recuperen y presenten información de una base de datos sin decirle exactamente las modificaciones que desea realizar en la base de datos (en forma de consultas), ADO proporciona el objeto **Recordset**.

Una vez establecida la conexión por medio del objeto Connection se utilizan los objetos Recordset, para enviar consultas y recuperar datos, respectivamente. El objeto Recordset tiene las funciones necesarias para recuperar y presentar un conjunto de filas, o registros, de una base de datos (dependiendo de las restricciones de las consultas).

Un objeto Recordset está compuesto a su vez de objetos Field, cada uno de los cuales contiene el valor del campo de la tabla recuperado en la consulta.

Cuando se ha terminado con la BD, se cierran todos los objetos abiertos, utilizando su método Close, y se establecen todos a Nothing para liberar la memoria reservada.

Código interno de la pagina "gl_salidapaquete.asp" que recibe y procesa la información.

```

<%@LANGUAGE="VBSCRIPT"%>      LENGUAJE DE DESARROLLO PARA LA PAGINA ASP.
<!--#include file="Connections/Cnx.asp" -->  PAGINA DE CONEXION A LA BASE DE DATOS
                                             A TRAVES DE UN DSN.

<%

Dim CodCD, NumSerie, ID_Paquete, Fecha      DECLARACION DE VARIABLES QUE RECIBEN
                                             LA INFORMACIÓN A PROCESAR

ASIGNACIÓN DE LA INFORMACIÓN A LAS VARIABLES

CodCD = 0 + Request.QueryString("CodCD")
NumSerie = "" & Request.QueryString("NumSerie")
ID_Paquete = 0 + Request.QueryString("ID_Paquete")
Fecha = "" & Request.QueryString("Fecha")

RECORDSET PARA EL ACCESO A LA INFORMACIÓN

set RcSet = Server.CreateObject("ADODB.Recordset")
RcSet.ActiveConnection = MM_Cnx_STRING
RcSet.Source = "SELECT * FROM CDs WHERE CodCD = " & CodCD
RcSet.CursorType = 0
RcSet.CursorLocation = 2
RcSet.LockType = 3
RcSet.Open()
RcSet_numRows = 0

IF RcSet.BOF AND RcSet.EOF THEN
  RcSet.AddNew          CREA NUEVO REGISTRO
  RcSet.Fields("ID_Paquete").Value = ID_Paquete
  RcSet.Fields("NoSerie").Value = NumSerie
  RcSet.Fields("CodCD").Value = CodCD
  RcSet.Fields("Fecha").Value = Fecha
  RcSet.Update
END IF

RcSet.Close()
%>

```

Finalmente es importante mencionar que si la petición viene del Subsistema de Registro, obtención y consulta de licencias de uso en Internet, para la activación de un producto X vía internet, el sitio en internet se encarga de realizar el registro del usuario final (registro de datos personales nombre, dirección, teléfono, etc) y devolver la clave de acceso correspondiente a dicho producto, para que dicho sistema finalice su proceso de obtención de uso de software.

En el siguiente tema se describe mas a detalle este proceso, ya que este es parte del "Subsistema de registro, obtención y consulta de licencia de uso en Internet".

Subsistema de Registro, Obtención y Consulta de Licencias de Uso en Internet

Como se estableció en la etapa de análisis, este subsistema está constituido por un sitio Web en Internet y realiza una comunicación directa con el servidor de licencias de uso en Internet; mediante este sitio Web el usuario final del programa de aplicación, puede registrar su copia de software y obtener la licencia de uso correspondiente.

Este subsistema es la alternativa para la obtención de la licencia de uso y activación de software en caso de que el usuario final no disponga de una conexión a Internet en el equipo final de instalación. El usuario debe acceder a este sitio Web desde cualquier equipo con acceso a Internet, o bien puede contactar vía telefónica a su distribuidor para que este último sea quien acceda al sitio.

El sitio Web está integrado por un conjunto de páginas Web ASP, cada una de las cuales realiza una función específica del subsistema. A continuación se listan las funciones básicas a desarrollar para la creación del presente subsistema.

- Autenticación para acceso a sitio
 - Primer autenticación: Ingresar número de serie
 - Segunda autenticación: Ingresar código de usuario
- Registro de usuario
- Consulta de licencias de uso
- Obtención de licencia de uso
- Actualización de datos de registro

Cada una de estas funciones están desarrolladas mediante páginas Web conteniendo procedimientos y funciones escritos en lenguaje ASP que permiten una comunicación directa hacia la base de datos del servidor de licencias de uso.

Autenticación para acceso a sitio

La autenticación para el acceso al sitio, está determinada por el número de serie que acompaña a la copia del programa de aplicación y el código de usuario asignado después de su registro, por ello se ha desglosado en dos autenticaciones.

Primer autenticación: Ingresar número de serie

El usuario debe ingresar el número de serie de su paquete de software, la imagen siguiente muestra la página Web solicitando el número de serie:

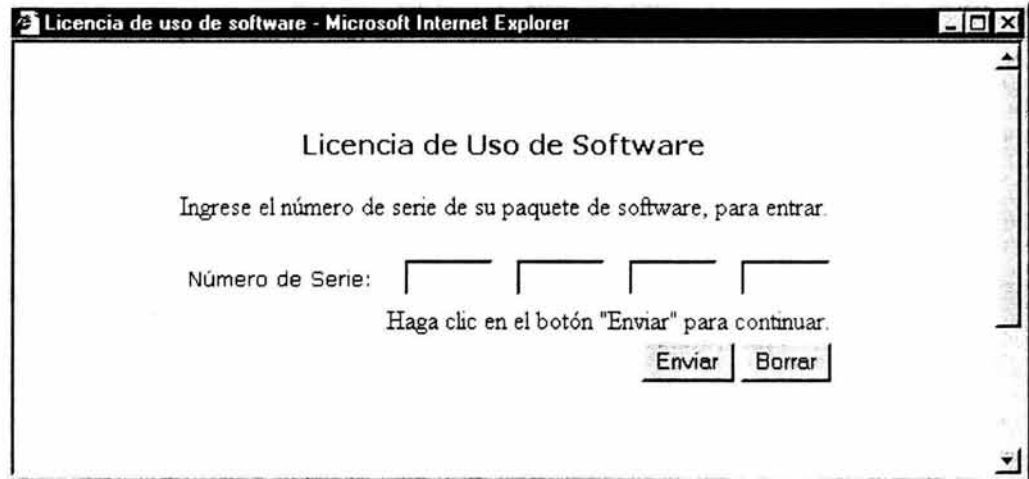


Figura IV.28 Primer autenticación

Como se observa en la figura, se solicita se ingrese el número de serie, para ello se cuenta con cuatro cajas de texto separadas mediante un guión y con una longitud máxima de cinco caracteres en cada una de ellas, correspondiendo con el formato del número de serie proporcionado.

Una vez ingresado el número de serie, el usuario debe hacer clic en el botón "Enviar" para que el sistema proceda con la validación y comprobación.

Validación de número de serie: Como se ha descrito en el apartado correspondiente al "Generador y administrador de licencias de uso" el número de serie esta conformado por 20 caracteres alfanuméricos, que contienen codificado un número entero aleatorio, número que identifica de forma única al paquete de software. Por consiguiente el proceso de validación consiste en obtener el número aleatorio.

A continuación se describe el algoritmo para comprobar la validez del número de serie:

- Recuperar el número de serie ingresado por el usuario.
- Comprobar su longitud de 20 caracteres. Una longitud menor lo invalida.

- Decodificar caracteres alfabéticos su correspondiente valor numérico según tabla de asignación utilizada para su creación.
- Recuperar los 10 dígitos correspondientes al número entero aleatorio y realizar cálculos inversos para decodificarlo. Al primer resultado incorrecto se determina la invalidez del número de serie.
- Recuperar los 10 dígitos restantes, los cuales constituyen un código de verificación, y realizar cálculos inversos para decodificarlo. Al primer resultado incorrecto se determina la invalidez del número de serie.

Si se ha validado exitosamente el número de serie ingresado por el usuario, debe de comprobarse que dicho número exista en el servidor de licencias de uso, de lo contrario, si el número no es válido se muestra una página indicando que el número de serie no es válido.

El proceso de comprobación de existencia en servidor de licencias es descrito a continuación.

Comprobación de número de serie: Se realiza una búsqueda, en la base de datos del servidor de licencias de uso, del número entero aleatorio obtenido para así comprobar su existencia:

- Si no existe el número en la base de datos del servidor, por lo tanto el número aún no se ha distribuido con algún paquete de software y se muestra una página indicando que el número de serie ingresado no es válido.
- Si se ha comprobado la existencia del número se procede a verificar si el número de serie ya había sido registrado; si aún no ha sido registrado, se muestra el formulario de registro, mas adelante se describe la función de registro de usuario. Si el número ya fue registrado se continua con la segunda autenticación, la cual se describe a continuación.

Segunda autenticación: Ingresar código de usuario

Si el número de serie ya ha sido registrado se muestra una nueva página solicitando se ingrese el código de usuario que le fue otorgado cuando se registro.

La figura siguiente muestra la página de solicitud de este código. La misma página realiza la búsqueda en la base de datos del servidor de licencias de uso y comprueba la existencia del código de usuario, después se procede a verificar si corresponde el código de usuario con el número de serie ya ingresado. Si el código

de usuario no existe o no corresponde con el número de serie, se muestra una nueva página indicando que el código no es válido.

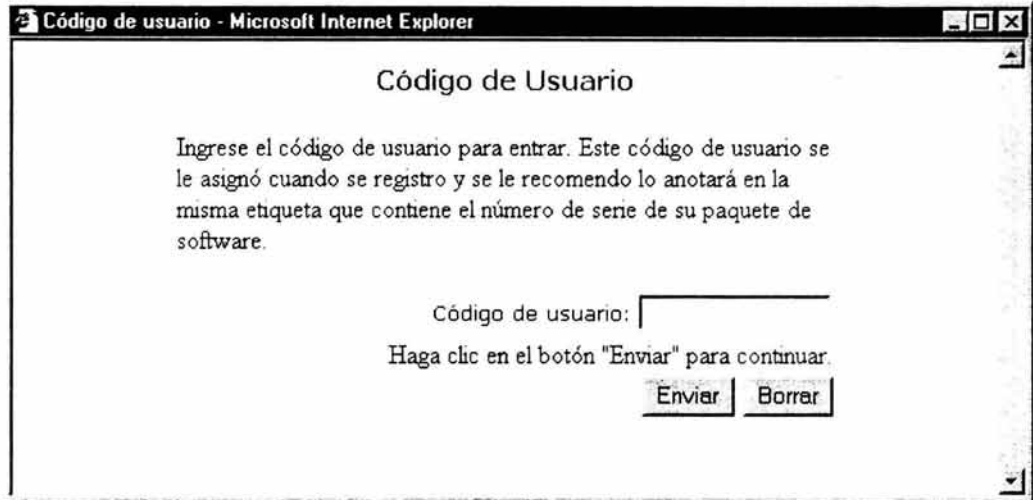


Figura IV.29 Segunda autenticación

Si se ha comprobado la existencia del código de usuario y su correspondencia con el número de serie ingresado, se crean variables de sesión que permiten se muestre la página de información de licencias de uso correspondientes al número de serie ingresado. La página recupera del servidor los datos de usuario, licencias disponibles, concedidas y permite el acceso a la página de obtención de licencia de uso, esta página es descrita más adelante.

Hasta este punto hemos expuesto el proceso de autenticación para el acceso al sitio. Se continua ahora con las siguientes funciones del subsistema.

Registro de usuario

Después de comprobar la existencia del número de serie en el servidor de licencias de uso, se verifica si el número ya fue registrado, de no estar registrado aún, se continua con el proceso de registro de usuario para el mismo número de serie.

El registro del usuario es a través de una nueva página Web conteniendo un formulario solicitando datos personales del usuario final que registra por primera y única vez el número de serie correspondiente al paquete de software.

Figura IV.30 Formulario de registro

La información ingresada en el formulario es validada y enviada a la base de datos del servidor, por la misma página. La información de registro es guardada en un nuevo registro de datos y se le asigna automáticamente un número entero aleatorio único.

El número entero aleatorio que constituye la llave única de identificación del usuario del número de serie ingresado y este número es asignado al número de serie dentro de la misma base de datos en la tabla de datos correspondiente. Además es considerado como el "Código de Usuario" asignado, es único, es decir, no existe duplicidad y esta relacionado únicamente con sólo un número de serie.

Después de que el usuario se registra, se le asigna el código de usuario y este se muestra en una nueva página, para que el usuario lo anote en la misma etiqueta que contiene el número de serie de su producto, a continuación se muestra un ejemplo de un código de usuario asignado después del registro.

Figura IV.31 Código de usuario asignado

Código de usuario

El código de usuario es el identificador del dueño único de un número de serie registrado. Consiste de un número entero en el intervalo de -2147483648 a 2147483648, para valores negativos se le antecede una letra "A" mayúscula, en representación del signo negativo, por ejemplo un código de usuario podría ser: "A312564", que corresponde al número -312564.

El número que representa el código de usuario, es generado de forma aleatoria por el servidor de licencias de uso en Internet, justo en el momento que el usuario del software de aplicación registra su copia; como ya se ha comentado es único, y esta relacionado únicamente con solo un número de serie. Cada número de serie tiene solamente un dueño, lo cual quiere decir que puede ser registrado solo una vez.

Consulta de licencias de uso

Después de la autenticación de usuario, se muestra la página de consulta principal. En esta página se muestran los datos del usuario, las licencias de uso disponibles y concedidas.

Licencias de uso concedidas:

Las licencias de uso concedidas al usuario, están listadas en una tabla indicando el nombre del programa de aplicación y la fecha en que fue concedida, además se muestra una tabla listando las licencias de uso concedidas por reposición, es decir, aquellas licencias que dieron debido a que el usuario requirió de reinstalar su software por pérdida de la instalación previa.

Licencias de uso disponibles:

Las licencias de uso disponibles están listadas en una tabla indicando el nombre del programa de aplicación y contiene una liga para su obtención, además se muestra una tabla listando las licencias de uso disponibles por reposición. A continuación se muestra una página de ejemplo.

La figura siguiente muestra un ejemplo de esta página de consulta de licencias de uso. En ella se lista una licencia de uso disponible.



Figura IV.32 Consulta de licencias de uso

Como se observa en la figura, el usuario visualiza en pantalla la información correspondiente a datos del software de aplicación, datos de registro, lista licencias de uso concedidas y disponibles. Internamente la página esta realizando la consulta hacia la base de datos del servidor, tomando como criterios de selección: el número de serie ingresado y el código de usuario. La lista de licencias de uso disponible contiene una liga para acceder a la página Web que permite su obtención, a continuación se describe el proceso de obtención de licencia de uso.

Obtención de licencia de uso

Como se comento en un principio este subsistema constituye la alternativa para la activación de software, en el caso de que el usuario final no disponga de una conexión a Internet en el equipo final de instalación. De acuerdo al diseño en el subsistema "Asistente de registro y activación de software", éste genera un código de equipo y solicita se ingrese una clave de acceso que permite la activación del programa de aplicación; dicha clave de acceso puede obtenerse desde este sitio.

Para la obtención de la clave de acceso se debe hacer clic en la liga "Obtener de licencia de uso", ubicada en la lista de licencias de uso disponibles, se muestra una nueva página solicitando se ingrese el código de equipo que haya generado el "Asistente de registro y activación de software, la figura siguiente muestra la página que solicita el ingreso del código de equipo para la obtención de la clave de acceso.

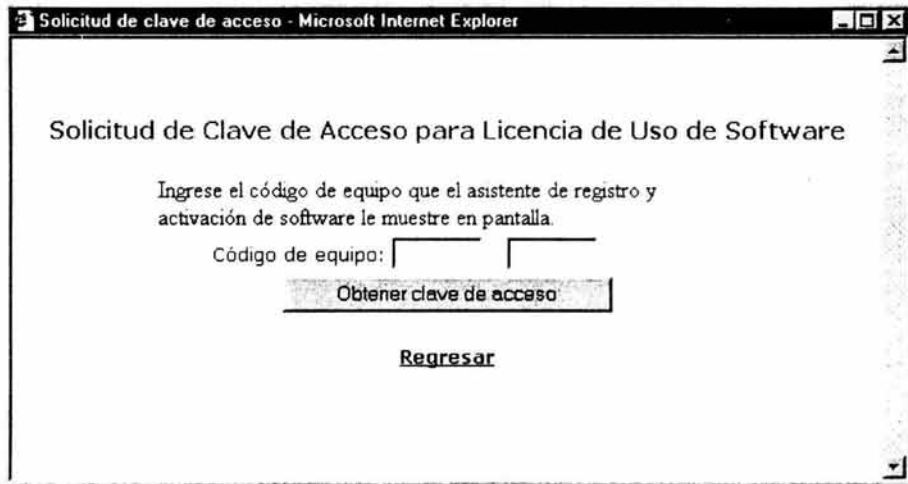


Figura IV.33 Solicitud de ingreso de código de equipo

La página se encarga de llamar a los procedimientos y funciones del servidor para validar el código de equipo ingresado. Como es sabido el código de equipo contiene codificado parte del número de serie del disco duro de la computadora donde se instaló el programa de aplicación, contiene el identificador de software y un código de verificación; por lo tanto la validación consiste en verificar su autenticidad a través del código de verificación y obtener el identificador de software.

Si se ha extraído exitosamente el identificador de software se consulta en la base de datos del servidor si el número de serie antes ingresado, dispone de una licencia de uso para un identificador de software igual al obtenido. De existir la licencia de uso para el software, se genera y muestra la clave de acceso correspondiente, para ser ingresada por el usuario en el "Asistente de registro y activación de software".

La clave de acceso es generada mediante un algoritmo que realiza la codificación de 10 valores numéricos generados mediante operaciones matemáticas entre los valores que conforman el código de equipo, con lo cual el mínimo cambio de uno de estos elementos lo invalidaría y además únicamente válido para el equipo de computo donde se genero el código de equipo. La clave de acceso esta conformada por dos series de cinco caracteres alfanuméricos, como se muestra a continuación:

XXXXX - XXXXX

Donde la "X" puede ser cualquier dígito del cero al nueve o una letra del alfabeto, es decir, cualquier letra entre la "A" y la "Z", exceptuando la letra "Ñ".

Considerando que cada carácter de la clave de acceso puede tomar 35 diferentes valores y son 10 caracteres que lo conforman, se tienen 35^{10} posibles combinaciones distintas, de las cuales solo una es válida para el código de equipo ingresado.

La imagen siguiente presenta un ejemplo de una clave de acceso generada para la obtención de una licencia de uso disponible.



Figura IV.34 Clave de acceso generada

Después de la obtención de la licencia de uso, al regresar a la página de consulta lista la licencia de uso antes disponible, ahora como concedida.

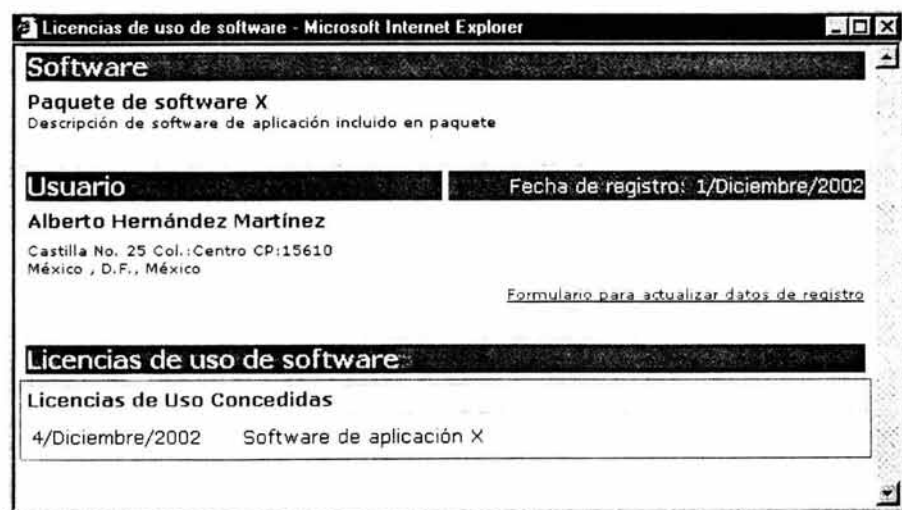


Figura IV.35 Después de obtener licencia de uso

A continuación se presenta el algoritmo para la generación de la clave de acceso, es esencial que en la codificación de la misma se emplee la misma tabla de asignación utilizada en la codificación del número de serie.

- Decodificación de los caracteres del código de equipo para la obtención de los valores numéricos.
- Generar 10 valores numéricos, mediante operaciones matemáticas entre los valores numéricos del código de equipo, pero con la restricción de que el resultado de la operación sea un número entero comprendido dentro del intervalo de 0 a 35.
- Codificar cada uno de los 10 valores numéricos generados, a su correspondiente dígito o letra según sea su valor y de acuerdo a la misma tabla de asignación utilizada en la generación del número de serie del software de aplicación.

Es importante mencionar que para disminuir la posibilidad de que pueda ser descifrada la clave de acceso, las operaciones matemáticas deben ser tan complejas como sea posible y además puede incrementarse la cantidad de caracteres que conforman la clave de acceso.

Actualización de datos de registro

En la misma página de consulta de licencias de uso se dispone de una liga a la página que contiene el formulario de actualización de datos de registro, en este formulario el usuario puede ingresar la información que haya cambiado, como puede ser, su domicilio o su dirección de correo electrónico.

Actualización de datos de registro - Microsoft Internet Explorer

Información de registro de usuario

Nombre(s): *	A. Paterno: *	A. Materno:
Alberto	Hernández	Martínez
Calle y Número:	Colonia:	CP:
Castilla No. 25	Centro	15610
Ciudad:	Edo. o Provincia:	País:
México	D.F.	México
Teléfono: (55)	55-44-33-22	ext.
E-mail: *	alberto@correo.com.mx	

Guardar Cambios

* Campos Obligatorios

Figura IV.36 Formulario de actualización de datos de registro

La información del formulario es enviada hacia la base de datos del servidor, posteriormente el subsistema generador y administrador de licencias de uso puede recuperar la información actualizada e ingresarla en su propia base de datos.

Hasta aquí, el desarrollo del sistema propuesto en este trabajo de tesis a concluido, a continuación en el siguiente capítulo se explica como se realiza la implantación del mismo y cual debe ser el mantenimiento que se tiene que llevar a cabo, para que el sistema sea 100% funcional y cumpla con las expectativas propuestas al inicio de este trabajo.

Implantación y mantenimiento del sistema

Pruebas

Implantación del sistema

Instalación

Capacitación a usuarios

Revisión del sistema

Mantenimiento del sistema

Durante la fase de desarrollo del sistema se realizaron para cada subsistema pruebas de ejecución necesarias para la detección y corrección de errores de codificación. De igual forma se realizaron pruebas de integración de los subsistemas para lograr su correcta comunicación, pero es imprescindible que antes de la implantación del sistema se realicen otro tipo de pruebas, las cuales no comprueban el software en sí, sino la eficiencia y confiabilidad en el funcionamiento del software en general. A continuación se describen las pruebas que se realizan antes de la implantación:

- **Pruebas de carga máxima:**

Existen tiempos críticos en muchos sistemas, particularmente en los sistemas en línea. Esta prueba sirve para determinar si el sistema soporta el volumen de actividades que ocurren cuando esté en el punto más alto de su demanda de procesamiento.

- **Pruebas de almacenamiento:**

La prueba requiere de almacenar información continuamente hasta que se alcanza la capacidad máxima. Al comprobar las capacidades ofrecidas y reales se verifica por un lado, la exactitud e integridad de la información y permite al mismo tiempo dar un juicio acerca de la capacidad real.

- **Pruebas en tiempo de ejecución:**

Durante las pruebas parciales, se usan conjuntos relativamente pequeños de datos para hallar errores o provocar fallas, por lo tanto, los usuarios detectan que tan rápido o lento es el sistema sólo hasta después de que ha sido instalado y cargado con grandes volúmenes de datos, esto puede ocurrir demasiado tarde. Sirven para determinar cuanto tiempo se lleva en recibir una respuesta a una consulta, hacer una copia de respaldo, mandar una petición y recibir una respuesta.

También incluye corridas de prueba para conocer el tiempo necesario de selección, ordenación, actualización y eliminación de grandes cantidades de información.

- **Pruebas de recuperación:**

Se debe suponer que el sistema fallará y que los datos se dañaran o perderán. Aunque se escriban planes y procedimientos para cubrir estas situaciones, también se deben probar mediante la creación de un evento de falla o pérdida de datos, en donde los usuarios se vean forzados a volver a cargar y recuperar la información a partir de una copia de respaldo.

- **Pruebas de seguridad:**

Se deben comprobar los procedimientos de autenticación y codificación de información para evitar el robo de información o accesos no permitidos al sistema.

- **Pruebas de facilidad de uso:**
Es de gran relevancia que el usuario final encuentre sencillo el sistema, aún sin poseer profundos conocimientos de computación, si el sistema posee facilidad de uso, el usuario lo asimilará rápidamente, hará un uso adecuado del mismo y lo aprovechará al máximo.
- **Pruebas de configuración e instalación:**
Consiste en probar el sistema en diferentes equipos de cómputo, con distintas configuraciones de software y hardware. También incluye pruebas de instalación y desinstalación.
- **Pruebas de regresión:**
Uso durante el mantenimiento del sistema cuando se hacen cambios o mejoras. Asegura que los cambios no afecten la funcionalidad actual de otros programas.

Después de comprobar la eficiencia y confiabilidad de uso del sistema se procede con su implantación.

Implantación del sistema

Como resultado del análisis y diseño se procede con la implantación del sistema desarrollado ya probado. La implantación consiste en un conjunto de procesos necesarios para poner en operación el sistema construido y permitir al usuario obtener beneficios por su funcionamiento.

Ahora bien, si no se realiza una implantación correcta, el sistema no cubrirá las necesidades y requerimientos para los que fue diseñado, esto puede traer como consecuencia la generación de desconfianza en los resultados del sistema, lo que se puede traducir en un limitado o nulo uso del sistema. Por el contrario una implantación exitosa generará confianza, un uso y aprovechamiento al 100% del sistema.

El proceso de implantación se divide en tres etapas:

- Instalación del sistema
- Capacitación de usuarios
- Revisión del sistema

A continuación son expuestas cada una de las etapas, enfocadas en el sistema de software del presente trabajo de tesis.

- **Instalación del sistema**

Se define la forma de instalación para cada uno de los subsistemas que integran al sistema global.

Subsistema Generador y administrador de licencias de uso de software

Este subsistema cuenta con un asistente de instalación, creado mediante el "Asistente de empaquetado y distribución", este asistente viene junto con Visual Basic, y permite generar un archivo de instalación de cualquier aplicación; recopilando todas las librerías y componentes de software necesarios para la operación del sistema a instalar. Para poder distribuir el archivo de instalación, este es almacenado en un disco compacto, de esta forma se puede ejecutar la instalación del sistema en el equipo donde se desea implantar, es importante mencionar que el equipo de destino debe cumplir con los mínimos requerimientos establecidos durante la etapa de planeación.

Al ejecutar el archivo de instalación, se inicia el asistente que guía al usuario en el proceso de instalación. Al concluir la instalación se han copiado y registrado los componentes de software necesarios, de igual forma se ha copiado en la carpeta de destino definida durante la instalación, el archivo ejecutable del subsistema, la base de datos y su correspondiente archivo de ayuda. El mismo asistente de instalación se encarga de generar el acceso directo, dentro del menú de inicio de Windows y en el escritorio del sistema, para iniciar la ejecución del subsistema.

Subsistema Asistente de activación de software

Al igual que el subsistema anterior, este cuenta con un archivo de instalación que se encarga de copiar y registrar el componente de software que constituye a este subsistema. Este componente es copiado dentro de la carpeta "System" o "System32" si el sistema operativo es Windows NT o 2000. También es copiado el archivo de ayuda correspondiente a este subsistema. Después de que el subsistema es registrado en el sistema, se puede hacer referencia al mismo para ser incluido en el entorno de diseño de Visual Basic y poder ser implantado en cualquier programa de aplicación a desarrollar.

En la siguiente figura, se hace referencia al componente de software "Asistente de Registro y Activación de Software", disponible en el sistema mediante el archivo "AsistRA.dll" localizado en la ruta: "C:\Windows\System". Teniendo la referencia al componente puede entonces implantarse en el programa de aplicación.

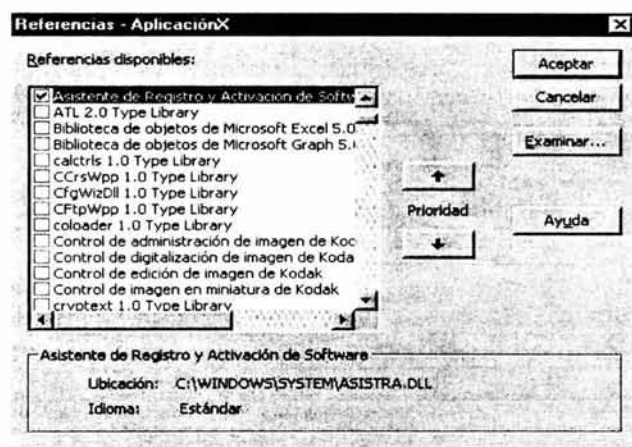


Figura V.1 Referenciado de componente de software.

La forma de implantar este subsistema, consiste en crear una instancia hacia el objeto contenido en el componente de software, asignar los valores necesarios a sus propiedades y ejecutar el método que se encarga de verificar la licencia de uso en el equipo del usuario final del programa de aplicación donde es implantado este subsistema. A continuación se muestra un ejemplo del código fuente de implantación.

```
Dim AsistRA As AsistRegistroActivacion
Set AsistRA = New AsistRegistroActivacion '//Creación del objeto
AsistRA.TituloLicencia = "Licencia de uso para software X"
AsistRA.NombreCia = "Cia. Desarrolladora de Software X"
AsistRA.UrlHttp = "http://www.empresa_x.com.mx/usuarios/"
AsistRA.IDSoftware = 2

If AsistRA.VerificaLicUso = False Then '//Comprobación
    End                               '// Fin de aplicación
End If

Set AsistRA = Nothing '// Eliminación del objeto
```

El código anterior debe ser llamado al inicio del programa de aplicación y este verifica la existencia de licencia de uso para el software, de no encontrarse muestra el asistente de registro y activación para la obtención de la licencia de uso del software que lleva por título "X", desarrollado por la compañía "X", la cual cuenta con su servidor de licencias de uso en la dirección de Internet "http://www.empresa_x.com.mx/usuarios/" y el identificador de software es "2".

En el ejemplo anterior, se termina la ejecución del programa de aplicación, si al verificar la licencia de uso, esta no es encontrada; queda a criterio de la empresa desarrolladora del programa de aplicación donde se implantará este subsistema, el hecho de que se finalice la ejecución o simplemente se restrinjan algunos módulos de la aplicación, si no se encontró la licencia de uso en el equipo del usuario final. Es decir, las instrucciones siguientes a la verificación de la licencia de uso, por parte del subsistema, están determinadas por el análisis y diseño que se haya hecho del programa de aplicación.

Servidor de licencias de uso en Internet

Subsistema de registro, obtención y consulta de licencia de uso en Internet

La implantación del “Servidor de licencias de uso en Internet” y el “Subsistema de registro, obtención y consulta de licencia de uso en Internet”, consiste en transferir el directorio que los contiene hacia el sitio en Internet perteneciente a la empresa desarrolladora de software. Previamente se ha establecido que el sitio debe dar soporte a páginas ASP.

Después de transferir el servidor, se procede con la creación de la fuente de recurso de datos (DSN), que permite realizar la conexión con la base datos, contenida en el mismo directorio que fue transferido al sitio de Internet. Para la creación del DSN debe accederse al panel de control del sitio Web.

• **Capacitación a usuarios**

Después de concluir la instalación y pruebas de funcionamiento se continúa con la capacitación de los usuarios para cada uno de los subsistemas. La capacitación consiste de un curso sobre el uso y manejo específico para cada subsistema dirigido al personal de la empresa desarrolladora de software.

Para la operación del sistema solo es necesaria una persona, la cual se encarga de operar el “Subsistema generador y administrador de licencias de uso” y el “Subsistema de registro, obtención y registro de licencias vía Internet”.

Esta persona tiene asignada la responsabilidad de llevar a través del primer subsistema, el registro de productos de software, registro de paquetes de software, generación de número de serie para cada paquete a distribuir y mediante el segundo subsistema proporcionar vía telefónica licencias de uso a los usuarios finales de los paquetes de software.

Existen dos aspectos para la capacitación de los usuarios: La familiarización con el sistema de procesamiento en sí y la posibilidad de llegar a situaciones

embarazosas que producirían en el usuario frustración, errores o ambos. Una buena documentación, aunque esencial, no reemplaza la capacitación. No hay sustituto para la operación directa del sistema, mientras se aprende la utilización de todas las herramientas del mismo.

La capacitación de usuarios se puede obtener de distintas formas. Las actividades de capacitación pueden llevarse a cabo en las instalaciones del proveedor, en locales rentados o en las instalaciones de la empresa donde se implantará el sistema.

La ventaja de ofrecer capacitación para el sistema en el lugar de la empresa, es que la instrucción se puede unir a la organización donde se ofrece y enfocarse a procedimientos especiales usados en ese marco, los planes de la organización para su crecimiento y cualquiera de los problemas que hayan surgido.

La capacitación debe complementarse con el uso de material de apoyo tal como materiales impresos o audiovisuales.

Sin embargo, no hay sustituto para la experiencia directa. Los manuales de capacitación son aceptables para la familiarización, pero las experiencias de usar realmente el sistema, corregir errores y hallar situaciones inesperadas son la mejor y más duradera forma de aprender. El enfoque de capacitación más común consiste en crear un caso de estudio que incluya todas las situaciones que se hallen frecuentemente, las cuales pueden ser manejadas por el sistema y que el usuario debe conocer. Los usuarios tienen que usar el sistema para llegar a las situaciones reales, es decir, introducir los datos en la forma pedida, procesarlos y preparar informes.

La capacitación debe asegurar que el usuario pueda manejar todas las operaciones posibles, tanto rutinarias como extraordinarias. Si el sistema necesita ser implantado en un nuevo equipo, la capacitación debe incluir aspectos básicos sobre como extraer la información del equipo actual, instalación del sistema en el nuevo equipo y restauración de información. El usuario debe aprender acerca de los desperfectos más comunes, como reconocerlos y que pasos llevar a cabo cuando ocurran. Como parte de la capacitación se debe proporcionar una lista de los posibles problemas y la forma de resolverlos, determinando si el problema que surge es causado por el equipo, el software o al utilizar el sistema.

La mayor parte de la capacitación de los usuarios tiene que ver con la operación del sistema en sí. Las actividades de manejo de datos que reciben la mayor atención en la capacitación, son la captura de datos, la edición de datos, la formulación de consultas y el borrado de registros de datos. Los usuarios deben

conocer los aspectos básicos sobre el uso de equipos de cómputo, el respaldo de información en disco, uso y manejo de impresoras.

Manual de usuario del sistema

La documentación del sistema esta disponible en medio impreso y en disco compacto. Independientemente de la capacitación obtenida, sirve como apoyo a los usuarios para resolver cualquier duda o aclaración que tengan en cuanto al funcionamiento y operación del sistema.

• **Revisión del sistema después de la implantación**

Después de implantar el sistema se hace una revisión por los usuarios y el encargado del proyecto. Esto no sólo es una práctica normal, sino que debe ser un proceso formal para determinar cómo esta funcionando el sistema, cómo ha sido aceptado y cuáles ajustes son necesarios. La revisión también es importante para recabar información para el mantenimiento del sistema. Puesto que ningún sistema es en realidad totalmente completo, el sistema permanecerá mientras no se requieran cambios debido a desarrollos internos como, nuevos requisitos legales, estándares de la industria o la competencia. La revisión después de la implantación es la primera fuente de información de los requisitos de mantenimiento.

Prueba de procedimientos

Los manuales de documentación y ejecución que dicen al usuario como llevar a cabo ciertas funciones se prueban fácilmente pidiendo al usuario que los siga en forma exacta por medio de una serie de eventos. Es sorprendente como pueden surgir preguntas ante el hecho de no incluir instrucciones sobre cuando oprimir la tecla ENTER, cuando quitar los discos o que hacer cuando no se imprime un reporte. Es común que las descripciones de los detalles no estén dentro de la documentación, este tipo de prueba no sólo muestra donde se necesitan, sino también en que lugar están equivocados, es decir, donde las acciones sugeridas en la documentación no son compatibles con las que realmente hay que llevar a cabo para hacer que el sistema funcione.

Finalmente después de implantar el sistema, capacitar a los usuarios y completar la puesta a punto se debe hacer una revisión continua para determinar si el sistema cumple con las expectativas y donde son necesarias las mejoras. La revisión y retroalimentación de información con los usuarios finales no solo indica, que tan bien está diseñado e implantado el sistema sino que también es una fuente valiosa de información para la fase de mantenimiento del sistema, la cual se aborda a continuación.

Mantenimiento del sistema

Generalmente la actividad de mantenimiento es bastante imprevista ya que se necesita de un periodo de observación del comportamiento del sistema en operación real, para determinar si se requiere corregir algún error, optimizar algún proceso, incluir alguna función no prevista por el usuario, o bien prevenir algunas fallas futuras.

Cuando se entrega un sistema se piensa que hasta ahí se trabajará con el producto entregado, sin embargo, se continúa con una actividad encargada de mejorar, adaptar y corregir el sistema, es decir, el sistema se desarrolla y posteriormente pasa a una evolución continua incrementando la numeración de su versión original; cualquier proceso que evoluciona tiende a dirigirse a dos caminos, se vuelve obsoleto por no recibir el mantenimiento oportuno ó sufre modificaciones durante casi toda su vida.

Cuando se descubre un error o se desea realizar una mejora en el sistema, algunas veces se realiza de manera inmediata (dependiendo de la urgencia), o se realizan las modificaciones con base a un calendario de actividades paralelas con otros proyectos, ó simplemente no se hace nada (debido a la falta de prioridad del proyecto o a la falta de recursos), por lo que el fin que se desea alcanzar en el proceso, es el de mejorar la facilidad con la que se puedan adaptar los cambios y reducir la cantidad de esfuerzo empleado en el mantenimiento. Algunas veces el error es tan grave, que se tienen que abandonar las actividades actuales para corregir de inmediato el problema, sin la evaluación de los posibles efectos secundarios y sin la adecuada actualización en la documentación, sin embargo estas actividades no se eliminan, sino que se llevan a cabo posteriormente, hasta que se haya corregido el problema.

Una vez que se haya resuelto la crisis se debe analizar si los cambios realizados no afectan a otros procedimientos del programa, lo cual pudiera generar incluso errores más serios. Este modo de actuación inmediata debe representar un porcentaje mínimo en la vida diaria y queda reservado exclusivamente para situaciones de verdaderas crisis. Existen cinco tipos de mantenimiento:

- **Mantenimiento correctivo:**

Son aquellos cambios precisos para corregir errores. Este mantenimiento inicia cuando se descubren errores y el sistema ya se ha puesto en marcha.

- **Mantenimiento adaptativo:**

Son las modificaciones necesarias cuando el entorno externo exige cambios de adaptación, por ejemplo, nuevos sistemas operativos, nuevas versiones

del lenguaje de programación, mejora en periféricos, etc., en el caso del sistema propuesto, es conveniente desarrollar los subsistemas en un lenguaje multiplataforma como es el caso del lenguaje JAVA, que permite el desarrollo de aplicaciones para distintos sistemas operativos.

- **Mantenimiento evolutivo:**

Son las incorporaciones, modificaciones y eliminaciones necesarias en el sistema para cubrir la expansión o cambio en las necesidades del usuario. Se produce cuando un sistema es utilizado con éxito y después se solicitan nuevas implementaciones y mejoras por parte de los usuarios, quienes retroalimentan al personal de informática, para la elaboración de dichos cambios, en el caso del sistema propuesto, este debe estar abierto a posibles cambios en la información recopilada del usuario final, como pueden ser campos adicionales relacionados con sus datos personales.

- **Mantenimiento perfectivo:**

Son las acciones llevadas a cabo para mejorar la calidad interna del sistema de software en cualquiera de sus aspectos: reestructuración del código, definición más clara del sistema y optimización del rendimiento y eficiencia.

- **Mantenimiento preventivo:**

Son las acciones necesarias para mejorar la facilidad de mantenimiento del sistema o proveer mejores bases para un mejor desempeño. Consiste en que antes de que se origine algún cambio, se analice y documente el proyecto de manera general y se tenga de esta forma un mejor control para las futuras modificaciones, en el caso del sistema propuesto este mantenimiento se enfoca al estudio y refinamiento de los algoritmos de generación de códigos para evitar el ataque de posibles piratas informáticos.

Muchas veces la calidad del mantenimiento y la cantidad de esfuerzo dedicado depende de si existe una documentación completa de proyecto, pues a veces, si solo se dispone del código fuente la actividad de mantenimiento comienza con un pesado análisis del código y quizá son mal interpretadas o difíciles de descubrir las estructuras de datos globales, las interfases de sistema, el rendimiento del sistema, la estructura del programa, etc.

También es imposible realizar las pruebas de regresión y en ellas observar si las modificaciones afectaron a otros procesos, pues no se cuenta con los registros de las pruebas anteriores. En cambio si en el proyecto a otorgarle mantenimiento, se llevo una metodología de software bien definida, entonces se comienza con una

evaluación de la documentación del diseño, se determinan las características estructurales de rendimiento y de interfaz, se desarrolla nuevo código fuente y se realizan pruebas de regresión mediante los registros contenidos en las especificaciones de pruebas, por lo que se puede decir que se mejora la calidad general del cambio y se reduce la cantidad de esfuerzo requerido.

Algunas veces los costos de mantenimiento representan aproximadamente el 60% de los gastos de todo el proyecto, y también se presentan costos intangibles de oportunidad, los cuales se generan cuando se pospone o suspende la oportunidad de un nuevo desarrollo debido a que los recursos se encuentran asignados al mantenimiento de un proyecto, es decir la productividad se ve seriamente afectada de modo que ni los desarrolladores, ni los usuarios finales se encuentran 100% satisfechos.

Otros costos intangibles son los siguientes:

- Existen clientes descontentos cuando realizan una petición de modificación al sistema, y ésta no se realiza en el tiempo esperado.
- Existe una disminución en la calidad global del desarrollo debido a los nuevos errores que introducen los cambios por mantenimiento.

Muchas veces no se considera la planeación de las actividades de mantenimiento y no se tiene la visión de que éstas empiezan desde el análisis, diseño y desarrollo del sistema, pues en el análisis se señalan estándares y principios generales para el proyecto de modo que se garantiza la uniformidad de los productos: durante el diseño estructural se debe recalcar la necesidad de realizar procesos claros, modulares y fáciles de mantener y “no” introducir tanto a los criterios de eficiencia y minimización en el espacio de memoria (en los casos que no sea necesario). En la codificación se deben de utilizar construcciones de una sola entrada y una sola salida, se debe adoptar un estilo de codificación simple, finalmente el mantenimiento se debe apoyar en toda la documentación técnica y del usuario elaborada durante el desarrollo del sistema.

La actividad de mantenimiento generalmente la realiza el mismo grupo que desarrollo la aplicación, por lo que ellos se encuentran familiarizados con el sistema, entienden la filosofía de desarrollo y saben sin problemas su funcionamiento, sin embargo cuando se trabaja en un nuevo proyecto entonces se pueden presentar interrupciones a surgir un nuevo mantenimiento. Cuando el mantenimiento lo va a proporcionar un grupo diferente al de desarrollo, entonces se debe tener especial cuidado en los estándares y en la documentación de alta calidad.

En muchas ocasiones existe una mejor recompensa y un mayor reto el desarrollar software nuevo que el darle mantenimiento a un software existente, por lo que la mayoría de los programadores desea desarrollar nuevas aplicaciones y no le interesa realizar mantenimientos, por lo que una forma de contrarrestar este problema puede consistir en rotar a los programadores entre el desarrollo, con esto se tiene la necesidad de hacer código de alta calidad y tener la adecuada documentación, se proporciona un mejor sentido de apreciación de las habilidades requeridas para ambas clases de programadores, se tiene más flexibilidad en el personal y se mejora la experiencia del mismo.

La solicitud de cambios es el primer paso para la administración del proceso de mantenimiento y sirve para solicitar correcciones o hacer adaptaciones al sistema. Si las adaptaciones al sistema son mayores, entonces se consideran como un proyecto nuevo. Esta solicitud es revisada por el analista y algunas veces por un comité que forma una junta de control de cambios, quienes deciden las acciones a tomar para la solicitud en proceso.

Las medidas de complejidad basadas en las propiedades del código fuente se pueden utilizar para comparar dos versiones de un programa, antes y después de una modificación y verificar si las actividades de mantenimiento están o no degradando al código original, pues si la complejidad del código fuente se incrementa con cada modificación se puede llegar a tener un código fuente inmantenible a diferencia del código estructurado y fácil de comprender que se tenía de inicio.

El esfuerzo dedicado al mantenimiento de software se divide en actividades productivas (análisis y evaluación, modificación de diseño, codificación, realización de pruebas de regresión, etc.), y en actividades menos productivas (revisar el código fuente para comprender lo que hace, tratar de interpretar las estructuras de datos, las interfases, los límites de rendimiento, etc.).

Finalmente, existen factores que pueden dañar el archivo de base de datos. Los síntomas de una base de datos dañada pueden variar desde que aparezca “#Eliminado” en algunos registros hasta que no puedan abrirse los objetos de la base de datos o el archivo en Access. Las utilidades Reparar y Compactar de Microsoft Access son herramientas útiles para recuperar y optimizar archivos de base de datos de Microsoft Access. Para evitar que la base de datos sufra daños se tiene que realizar el siguiente mantenimiento:

1) **Compactar base de datos:** con esto se elimina los residuos de la misma. La operación puede tardar unos instantes dependiendo del tamaño actual del archivo. Es conveniente realizar esta operación cuando note que el sistema funciona muy lento.

2) **Reparar base de datos:** verifica y repara el archivo en caso que resultara dañado por corte de energía o algún problema en su computadora. La operación puede tardar unos instantes dependiendo del tamaño actual del archivo.

3) **Hacer copias de seguridad:** debe realizarlas el usuario cada vez que lo crea conveniente. De todos modos, el sistema realiza automáticamente una copia diaria de la base de datos.

Hasta este punto se ha completado el ciclo de vida del presente sistema de software.

Conclusiones

Conclusiones

La piratería existe debido a algo tan simple como los mecanismos de la **oferta** y la **demanda** en los que se basa nuestra economía capitalista. **La oferta existente** (programas con un precio de venta al público que habitualmente supera los \$2,000 pesos) **no concuerda con la demanda** (un producto de calidad a un precio accesible o cuando menos, razonable), lo que provoca la aparición de un **nuevo mercado** (el de la piratería) cuya oferta está más ajustada a la demanda, lo que satisface tanto a los productores de software pirata como a los consumidores. Dado que la oferta de este mercado rebaja tanto la **calidad** del producto (inexistencia de manual y soporte técnico) como su **precio** (reducido a casi la décima parte), cabe señalar la siguiente conclusión a la que llega la demanda:

“El producto original es tan caro que no importa esa pérdida de calidad, si se puede obtener a un precio razonable”.

Por otra parte el precio “elevado” de un producto de software original es justificable por los costos que implica su desarrollo y soporte técnico, por lo que la protección de los derechos de propiedad intelectual reporta beneficios económicos a toda nuestra sociedad. Al eliminar la piratería de software, la economía de nuestro país crece, ya que se estimula el crecimiento de la industria local en el campo de la informática, lo cual aumentará el número de nuevos empleos y mejorará la contribución fiscal para el Estado.

La piratería de software es un delito que nos puede llevar a afrontar serias consecuencias, por lo que es necesario enfocar los recursos adecuados y diseñar mecanismos que permitan en un futuro erradicar por completo el problema; es por ello que cómo egresados de la carrera de ingeniería en computación, comenzamos a asumir nuestro papel de ingenieros al implementar el sistema propuesto y del cual podemos concluir lo siguiente:

- Se ha creado un sistema integral que brinda una opción más para la protección de software, dicho sistema no pretende ser la solución total al problema de la piratería de software, pero sí una alternativa para su disminución.
- El sistema cumple con su misión de activar solo software registrado por un usuario que ha pagado por el mismo.

- El sistema es actualmente utilizado por una empresa de desarrollo de software, con lo cual se ha pasado de una simple propuesta a una aplicación real ya en uso.
- La importancia de Internet fue significativa para el cumplimiento del objetivo del sistema, de tal forma que constituye la vía de comunicación para la activación del software sin la limitante de horario y lugar de residencia del usuario final.
- Se han puesto en práctica algunos de los fundamentos teóricos obtenidos en el aula de clases, como es el caso de la metodología de análisis estructurada para el diseño de sistemas informáticos.
- El estudio preliminar de la presente tesis se fundamenta en una base de información presentada a lo largo de la carrera profesional, por lo cual, estamos convencidos que hemos realizado un sistema de acuerdo a las expectativas y herramientas disponibles en la actualidad.

Este tema fue una idea de acuerdo a la necesidad de brindar una opción más de protección de software, en contra de la piratería.

Aportación

La propuesta de diseñar un sistema para atacar un problema no resuelto en su totalidad hasta la fecha, se puede considerar como aportación de esta tesis.

Dentro de la fase de desarrollo del sistema se describen los algoritmos para la generación de códigos, que permitirán a futuros investigadores su análisis para su perfeccionamiento, o ser la base, que de principio a nuevos algoritmos para la protección de software.

Este trabajo de tesis queda a disposición de consulta en la biblioteca de la escuela y de la universidad, aportando un conocimiento teórico práctico en el ámbito informático, deseando sirva de ejemplo y de utilidad para los estudiantes de generaciones actuales y futuras.

Investigación a futuro

En el transcurso de este trabajo han quedado abiertas una serie de líneas de trabajo que vamos a resumir a continuación.

- Migración de subsistemas a un lenguaje multiplataforma como es el caso de Java, que permite la posibilidad de crear aplicaciones para distintos sistemas operativos.
- Seguir el refinamiento de los algoritmos empleados para asegurar que no sean descifrados por algún pirata informático.
- El sistema completo ha sido validado con un uso extensivo en un entorno real de toma de datos, y funciona sin problema alguno, pero existen áreas donde el rendimiento necesita ser mejorado.

Creemos que la evaluación más crítica a corto plazo del sistema propuesto en este trabajo de tesis es su continua utilización como punto de partida para estudios posteriores de codificación y análisis.

Bibliografía

Bibliografía

LINKS DE CONSULTA

Ingeniería de software

<http://www.itcg.edu.mx/ingsoft/index.htm>

<http://www.monografias.com/trabajos11/anasiste/anasiste.shtml>

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

Definición de sistema operativo

http://www.xsii.com/tutorials/internet/6_2.html

<http://dmi.uib.es/~abasolo/foininf/2001-2002/capitulos/1-ntroduccion.html#1.8>

Windows

<http://www.nacionlinux.com/propietario.html>

Visual Basic

<http://www.comsto.org/lg/vb.htm>

Fundamentos básicos del derecho de autor

<http://www.copyright.gov/circs/circ1-espanol.html>

<http://www.hardings.cl/publications/copyright.html>

http://www.enel.net/gacetajudicial/1999/69/propiedad_intelectual.htm

Protección antipiratería

<http://ocu.ucr.ac.cr/media/a05-1-2000.htm>

<http://arcade.ya.com/elabandon/reportajes/proteccion.html>

<http://www.stormymondays.com/biblioteca/pirateria.htm>

<http://www.bsa.org/es/>

Piratería en internet

http://www.asturpyme.com/noticiario_18b.htm

http://www.laempresa.net/noticias/2000/0007/20000726_13.htm

<http://www.udc.es/uep/p58.htm>

Internet como medio de comunicación

http://members.tripod.com/e_soule/tesis/inicio.html

Ciclo de vida

<http://www.andrew.cmu.edu/user/conzalez/Teaching/ISW1/Lifecycle.html>

<http://www.geocities.com/Athens/Olympus/8740/Ciclo.htm>

<http://www.uag.mx/66/Casca.htm>

<http://alarcos.inf-cr.uclm.es/doc/ISOFTWARE/isoftware.htm>

<http://nuyoo.utm.mx/~caff/doc/IntroIngdeSoftwareITAO.pdf>

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

Base de datos

http://atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema3_1.htm

http://atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema2_5.htm

<http://www.medal.org.ar/stadhelp/Std00020.htm>

<http://jtorres.mor.itesm.mx/bdt/BaseDatos.htm>

<http://fismat.umich.mx/~emurguia/mipagina/tesis/node34.html>

<http://www.monografias.com/trabajos11/colum/colum.shtml>

Base de datos relacional

<http://www.jegsworks.com/Lessons-sp/lesson1-2/lesson2-4database.htm>

<http://lucas.hispalinux.es/Postgresql-es/web/navegable/tutorial/sql.html>

<http://www.dcc.uchile.cl/~luguerre/cc20a/e-r.html>

<http://www.medal.org.ar/stadhelp/Std00020.htm>

<http://www3.uji.es/~mmarques/f47/apun/node1.html>

Diccionario de Datos

http://148.204.45.136:9000/sqlmx/paginas/documentacion/dic_datos/Datadict.html

Servidor Seguro

<http://www.puntaweb.com/crs/ssl.htm>

http://pyme.net.uy/documentos/sitio_seguro.htm

<http://www.uruguaytotal.com/videos/secure.htm>

<http://www.arsys.es/productos/alojamiento/svcommerce/sseguro.htm>

ASP

<http://www.ultrasist.com.mx/tecnologias/asp.htm>
<http://support.activeisp.es/web/database/dsn.asp>
<http://www.asptutor.com/asp/vart.asp?id=3>
http://www.programacion.com/asp/tutorial.asp_basics.html

ODBC

<http://www.iec.csic.es/criptonomicon/bd/accesobd.html>
<http://www.sqlmax.com/asp2.asp>
<http://www.dream-es.com/Content/Pages/dsn.asp>

Cliente-Servidor

<http://www.geocities.com/SiliconValley/8195/noscs.html>
<http://www.linti.unlp.edu.ar/trabajos/tesisDeGrado/tutorial/redes/clteserv.htm>

LAN

<http://www.cybercursos.net/cursos-online/lan/Wanmanlan.htm>

Análisis estructurado

<http://www.cs.us.es/cursos/bd/temas/disenio.html>
http://grulla.hispalinux.es/enunciados/analisis_estructurado.pdf
<http://www.itcg.edu.mx/ingsoft/herramgraf.htm>

ADO

<http://ainsuca.javeriana.edu.co/~cbustaca/Libros/ADOO.pdf>
<http://www.asptutor.com/asp/vart.asp?id=1>

LIBROS DE CONSULTA

"Análisis estructurado moderno"

Edward Yourdon
Ed. Prentice Hall, 1993

"Análisis y diseño de sistemas de información"

Whitten, Bentley & Barlow
Ed. IRWIN, 1996

"Ingeniería del software (Un enfoque práctico)"

Roger S. Pressman
Editorial McGraw Hill, Segunda edición 1990