Universidad de las Ciencias Informáticas

Facultad 7

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Estrategia de protección y licenciamiento para los software del Departamento de Software Médico Imagenológico del CESIM

Autora: Danayti Lorenmys Jerez Pompa

Tutores: Ing. Yoel González Mesa

Ing. Lázaro González Rodríguez

Ciudad de La Habana, junio del 2010

"Año52 de la Revolución"

DATOS DE CONTACTO

Tutores:

***Ing. Yoel González Mesa*:** Graduado de Ingeniero en Ciencias Informáticas en la Universidad de Ciencias Informáticas (UCI) en el año 2009. Posee la categoría de instructor recién graduado. Pertenece al Departamento de Software Médico Imagenológico del Centro de Informática Médica (CESIM) perteneciente a la Facultad 7. Se desempeña como Jefe del Área de Soporte, Investigación y Licenciamiento. Correo electrónico: ygmesa@uci.cu

Ing. Lázaro González Rodríguez: Profesor instructor, graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2007. Imparte la asignatura de Práctica Profesional 2. Actualmente se desempeña como Arquitecto del Departamento de Software Médico Imagenológico del Centro de Informática Médica. Correo electrónico: lgonzalezr@uci.cu

RESUMEN

En el año 2009 el Grupo de Procesamiento de Imágenes y Señales (GPI) cambia su nombre, convirtiéndose así en el Departamento de Software Médico Imagenológico, perteneciente al Centro de Informática Médica (CESIM). En este departamento se han desarrollado soluciones que actualmente son desplegadas en varios países, reportando grandes ganancias a la economía del país. Una de ellas es el producto alas PACS-alas RIS.

Esta y otras soluciones han sido desarrolladas en modalidades de aplicación web y escritorio para las plataformas Windows y Linux. Por la importancia que tiene su comercialización como software propietario, es que la dirección del departamento decidió realizar una estrategia para garantizar la protección de las mismas.

Este trabajo de diploma tiene como objetivo proponer una estrategia para el licenciamiento de los software desarrollados por el Departamento de Software Médico Imagenológico del Centro de Informática Médica (CESIM).

Como resultado se diseña y describe cada uno de los pasos de la estrategia, definidos en dos fases fundamentales así como también se tratan temas que puedan crear dudas a la hora de lograr un mejor entendimiento de la misma. También se generó como artefacto de salida una plantilla que servirá como guía para la aplicación de la misma.

PALABRAS CLAVES: estrategia, protección, proceso, software, desarrollo.

TABLA DE CONTENIDOS

INTRODUCCIÓN

En los últimos treinta años se han generado vertiginosos cambios en el desarrollo de la ciencia a nivel mundial, teniéndose como característica principal de esta etapa la rapidez con que los nuevos conocimientos son llevados a la práctica, es decir, que el conocimiento científico modifica la sociedad al convertirse en un producto concreto. En la actualidad, la humanidad depende cada vez más de los progresos de la ciencia.

Es a partir del último decenio del siglo XX cuando comienza a marcarse una convergencia entre algunas ciencias como son la computación, las telecomunicaciones, la electrónica y los medios de comunicación masiva. Esto propició que fueran desapareciendo las fronteras entre dichas ciencias, surgiendo así nuevos conceptos tales como: Tecnologías de Información, Sociedad del conocimiento, Telemática, Informática, etc.

Dichas ciencias en su conjunto son conocidas como Tecnologías de la Información y las Comunicaciones (TIC), las cuales no son más que el conjunto de recursos necesarios para manipular la información y entre los que se pueden encontrar por ejemplo, los ordenadores, los programas informáticos y las redes necesarias para convertir, almacenar, administrar, transmitir y encontrar dicha información, es por ello que, a medida que se han ido desarrollando las TIC, la sociedad se ha beneficiado de muchas maneras y en diferentes ramas, como la educación, el deporte, la salud, etc., facilitando así la vida de los habitantes del planeta.

Paralelo al empleo de las TIC en todos los procesos de desarrollo de la humanidad, han surgido peligros a la hora de proteger los productos que se obtienen de las mismas. Principalmente en la industria de software las últimas tendencias indican que la protección y el licenciamiento de las soluciones que se van desarrollando tienen gran importancia, ya que el uso no autorizado de las mismas puede provocar grandes pérdidas.

Se conoce que el licenciamiento de un software otorga el derecho legal de ejecutarlo y utilizarlo. Es una especie de contrato que regula el uso de la licencia del software. Normalmente, los contratos de licenciamiento permiten que el software sea ejecutado en un número limitado de ordenadores y que se realicen copias sólo con propósitos de respaldo. Actualmente se utiliza ese método de protección para evitar amenazas como la piratería que se ha vuelto uno de los principales riesgos en la industria. Esto posibilita

que se logre garantizar la seguridad de los productos y así mantener la confiabilidad del cliente y del mercado. (1)

Los software que se desarrollan en las empresas suponen un esfuerzo de equipo que combina las ideas y los talentos de programadores, redactores y diseñadores gráficos. Y, al igual que la mayoría de las obras creativas, tales como libros, composiciones musicales y películas, están protegidos por las leyes de propiedad intelectual.

El uso ilegal de los mismos puede propiciar una mayor exposición a virus de software, discos dañados o software defectuoso. Puede además generar la falta de documentación o documentación inadecuada, la ausencia de garantías, la falta de apoyo técnico del producto disponible para los usuarios que cuentan con las licencias adecuadas, demás de la imposibilidad de acceder a las actualizaciones de software que se ofrecen a los usuarios que cuentan con las mismas.

Se conoce que las licencias están basadas en los derechos del autor y son la base jurídica del software que se desarrolla y se utiliza. Mediante ellas se protegen los productos de los usos abusivos u oportunistas y son además una forma eficaz de proteger las inversiones que se realizan.

En Cuba, una de las instituciones encargadas del desarrollo de software es la Universidad de la Ciencias Informáticas (UCI) que fue fundada por el Comandante en Jefe Fidel Castro en el año 2002.Esta universidad tiene como objetivos principales lograr la informatización del país y desarrollar la industria del software para contribuir al mejoramiento de la economía cubana.

En dicha universidad se encuentra ubicado el Departamento de Software Médico Imagenológico del Centro de Informática Médica (CESIM).En él se han desarrollado productos como alas PACS y alas RIS que actualmente son utilizados en países como Cuba y Venezuela.

El proceso de despliegue de estas soluciones es llevado a cabo por los miembros del departamento. Esto no significa que dejen de existir riesgos en la protección de las mismas ya que puede suceder que una vez instalados en las estaciones de trabajo sean craqueados o que por falta de integridad de algún miembro, el software sea distribuido de forma ilegal.

Por otra parte, el departamento no cuenta con un sistema fiable a la hora de otorgar las licencias de uso de los productos, es decir, no existe la forma de garantizar que el número de licencias solicitadas por los compradores sea el mismo que se otorgue y esto no genere problemas a la hora de la instalación.

Es por ello que no se han podido conformar los demos de prueba, pues no se cuenta con una licencia que respalde las condiciones que se requieren para la utilización de los mismos antes de que los compradores se decidan a realizar los contratos.

La inexistencia de un método de protección que sea el más efectivo de acuerdo con las necesidades y características de los software que se desarrollan en el Departamento de Software Médico Imagenológico puede conllevar a que exista cualquier filtración o robo, lo cual derivaría en pérdidas millonarias para el CESIM y el país.

Después de analizar la situación antes expuesta se identificó como **Problema científico**: ¿Cómo facilitar la protección de los software que se desarrollan en el Departamento de Software Médico Imagenológico?

Para solucionar dicho problema el **Objeto de estudio** está dado por: el licenciamiento e software.

Delimitado en el **Campo de acción:** el licenciamiento de software en el Departamento de Software Médico Imagenológico.

Para dar solución a la problemática antes mencionada se ha definido como **Objetivo General:** Proponer una estrategia para el licenciamiento de los software desarrollados por el Departamento de Software Médico Imagenológico del Centro de Informática Médica (CESIM).

Para dar cumplimiento al objetivo antes mencionado se proponen las siguientes **Tareas de la investigación:**

1. Analizar el estado general sobre el tema:”licenciamiento y protección del software”.
2. Analizar los sistemas automatizados y los métodos existentes en la actualidad para el licenciamiento de software.
3. Elaborar una descripción detallada de cada uno de los métodos de licenciamiento de software.
4. Realizar una comparación entre los métodos de licenciamiento tanto de hardware como de software.

5. Evaluar las características de los software desarrollados por el Departamento de Software Médico Imagenológico del CESIM.
6. Valorar los sistemas y métodos más adecuados para proveer y activar una licencia en los software desarrollados por el Departamento de Software Médico Imagenológico del CESIM.
7. Desarrollar una estrategia para el licenciamiento de los software desarrollados por el Departamento de Software Médico Imagenológico del CESIM.
8. Probar las herramientas y tecnologías que se emplearán en la estrategia propuesta.
9. Validar las herramientas y tecnologías que se emplearán en la estrategia propuesta.

Para un mejor desarrollo de este trabajo se estructuró el documento en cuatro capítulos, los que se describen a continuación:

Capítulo 1. Fundamentación Teórica.

Incluye el estado del arte del tema licenciamiento y protección del software, a nivel internacional, nacional y en la Universidad; de las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad para la solución del problema que se enfrenta. Se expone un estudio crítico, valorativo, no una mera reproducción de referencias y estadísticas.

Capítulo 2. Características de los métodos de licenciamiento de software.

Incluye una descripción detallada de los métodos utilizadas para el licenciamiento de software, así como el análisis y la comparación entre los más empleados a nivel internacional.

Capítulo 3. Características de los software producidos por el Departamento de Software Médico Imagenológico del CESIM.

Incluye una descripción general de los software que se desarrollan en el Departamento de Software Médico Imagenológico del CESIM (alas PACS/alas RIS, etc.).Analizar características como: Lenguaje de desarrollo, Plataforma (Linux, Windows), si es una aplicación de escritorio o web, necesidades de protección, entre otras características que se consideren importantes para decidir el mejor método de protección.

Capítulo 4. Estrategia para el licenciamiento de los software desarrollados por el Departamento de Software Médico Imagenológico del CESIM.

Se define como tal, la estrategia más adecuada a emplear para la protección y el licenciamiento de los software del Departamento de Software Médico Imagenológico del CESIM. Se definen varias alternativas en dependencia del tipo de software. Se prueban las herramientas a utilizar.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

En este capítulo se abordará todo lo relacionado con el estado del arte de los métodos para proveer y activar una licencia de software, así como de las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad y que servirán de apoyo para la solución del problema que se enfrenta. Se argumentarán conceptos tales como: software, licenciamiento de software, licencias de software, derecho de autor, propiedad intelectual. Todo con el objetivo fundamental de brindar información complementaria para propiciar un mejor entendimiento de los capítulos venideros.

1.1. Licenciamiento de software.

El licenciamiento de software es el proceso de compra estatal de software legal, en tanto dicha autorización o licencia se requiera. Es un contrato que regula el seguimiento de las normas exactas que son descritas en la documentación que acompaña al producto, es decir, la licencia.

Hoy en día las soluciones de licenciamiento de software efectivas son aquellas que no sólo protegen el software contra la piratería sino que también disminuyen los costos operativos y permiten el incremento de ingresos a través de modelos de licenciamiento flexibles.

Para analizar este proceso se puede centrar la atención en dos campos fundamentales: el licenciamiento en Software Libre y el licenciamiento en el Software Propietario.

Del licenciamiento en el software libre se puede decir que es más económico. Ofrece mayor libertad de uso y distribución ya que las licencias de software libre existentes permiten la instalación del software tantas veces y en tantas máquinas como el usuario desee. Otorga una independencia tecnológica pues permite el acceso al código fuente lo cual facilita el desarrollo de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero. (2)

Sin embargo, existe la posibilidad de una generación más fácil de virus de software, dado que el código fuente también puede ser modificado con intenciones maliciosas. Este acceso libre al código permite que cualquier usuario pueda realizar con el software lo que estime conveniente ya que el distribuidor del mismo le otorga esa libertad en el contrato de licenciamiento. La consecuencia final es que falten piezas claves como son el caso de la documentación y que la evolución futura de los componentes del software no esté asegurada.

Por el contrario, en el licenciamiento del software propietario se prohíbe en alguna medida la copia, redistribución o modificación del software por parte de su propietario. Para realizar cualquiera de las acciones antes mencionadas se debe solicitar el permiso al dueño o pagar.

El proceso de licenciamiento en este tipo de software otorga la propiedad y decisión de uso del software a la empresa desarrolladora puesto que la competencia se podría apropiar inmediatamente del producto una vez finalizado, para sus propios fines.

Brinda un mejor acabado de la mayoría de aplicaciones ya que se nota de forma especial el cuidado y esfuerzo que ponen los desarrolladores en el software. Al fin y al cabo ganan dinero directamente con el producto y se debe demostrar porqué ese producto es la mejor opción. Por otra parte, se garantiza una mejor protección de las obras con copyright, ya que las soluciones son beneficiadas con mecanismos anti-copia.

1.2. Algunos conceptos básicos.

Software: equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

Licencia: es la autorización o permiso concedido por el titular del derecho de autor, en cualquier forma contractual, al usuario de un programa informático, para utilizar éste en una forma determinada y de conformidad con unas condiciones convenidas.

Derechos de autor: protegen la expresión de un contenido, no el contenido en sí mismo. Se desarrollaron para recompensar a los autores de libros o de arte. Las obras protegidas pueden expresar ideas, conocimientos o métodos libremente utilizables, pero se prohíbe reproducirlas sin permiso, total o parcialmente, con o sin modificaciones.

Propiedad intelectual: cualquier propiedad que, de común acuerdo, se considere de naturaleza intelectual y merecedora de protección, incluida las invenciones científicas y tecnológicas, las producciones literarias o artísticas, las marcas o identificadores de los productos, los dibujos, modelos industriales y las indicaciones geográficas.

1.3. Licenciamiento de software propietario. Métodos más usados.

En la mayor parte de las aplicaciones de software empleadas en la actualidad el licenciamiento se realiza bajo la modalidad de software propietario aunque no parece existir una estandarización de métodos para realizar ese proceso.

La clasificación deriva del fabricante. Se suelen denominar Contrato de Licencia de Usuario Final o CLUF *(*EULA*)*. Para *aplicaciones cliente-servidor*, puede haber una licencia para el software servidor, y también *Licencias de Acceso de Cliente* o *CAL*, que dan permiso para que máquinas individuales accedan a una aplicación o recurso proporcionado por el software servidor.

Existen también opciones de *licencias por volumen*: mediante un contrato especial entre el proveedor y el cliente, se negocia la cantidad de ejemplares del programa que se autorizan, generalmente dentro de las instalaciones de una empresa grande con cientos o miles de máquinas.

A continuación se argumentará más sobre algunos métodos de licenciamiento existentes para el software propietario.

1.4.1 Contrato de licencia de Usuario Final (CLUF).

El Contrato de Licencia de Usuario Final (CLUF) es el contrato entre el fabricante de la PC y el usuario final. Mayormente aparece ante los usuarios en la caja del software impreso en papel o como una ventana emergente que aparece cuando se va a instalar el software. Generalmente se deben aceptar los términos y condiciones de la licencia del software antes de instalarlo. Constituyen el método de licenciamiento de Microsoft con que más familiarizados están los usuarios actualmente. (3)

1.4.2 Licencias de Acceso de Cliente o CAL.

Son muy usadas en modelos basados en servidores, implica que los servicios se acceden al servidor desde una PC. Estas licencias se deben adquirir por cada terminal (PC, pager, PDA, etc.) accediendo a una aplicación en el back-office. A diferencia de una licencia cliente, donde en un modelo cliente/servidor se paga por el software y por el acceso a los servicios; en este caso, sólo se paga por el acceso ya que el software es proporcionado en forma gratuita o incluido en el terminal (por ejemplo: browsers).

Se utilizan como método de licenciamiento ya que son licencias que le dan al usuario el derecho a utilizar los servicios de un servidor. Estas licencias pueden resultar un poco confusas a la hora de utilizarlas por lo que se han definido tres secciones principales en las que estas licencias son aplicadas a los productos.

- ***Licenciamiento por "Usuario o Por Equipo".***

En este tipo de licenciamiento es necesario poseer una CAL de Windows individual para cada uno de los usuarios o equipos que acceden o utilizan el software de cualquiera de sus servidores. El número de CALs de Windows que se precisan equivale al número de usuarios o equipos que acceden al software del servidor.

Si se elige esta modalidad, la elección es permanente. Se puede, sin embargo, reasignar el CAL de Windows de un equipo a otro o de un usuario a otro, siempre y cuando la reasignación se haga o bien de forma permanente al eliminar un equipo o usuario determinado o de forma temporal al trasladar la utilización del CAL de Windows a un equipo de sustitución, mientras el equipo permanente se encuentra fuera de servicio o a un empleado temporal, mientras el trabajador habitual se encuentra ausente.

La modalidad Por Usuario/Por Equipo tiende a ser la designación de CAL de Windows más económica en los entornos informáticos gestionados, dónde múltiples servidores dentro de una empresa proveen servicios a la mayoría de equipos o usuarios.

- ***Licenciamiento por "servidor".***

Cuando se usa este método, un número específico de CAL se asocia a un servidor. El número de dispositivos que pueden acceder al servidor simultáneamente, de forma legal, está limitado en el contrato por el número de licencias adquiridas para ese servidor en particular. En instalaciones Por Servidor estas ultimas no están asociadas permanentemente con un dispositivo específico, si la organización añade otro servidor, y permanece instalado en modo Por Servidor, será necesario el uso de otras CAL para acceder al segundo servidor. Con instalaciones Por Servidor, el administrador de sistemas designa el número de dichas licencias que serán aplicadas al servidor durante el proceso de instalación, dependiendo del número que haya sido adquirido para ese servidor.

También se pueden licenciar servidores que estén disponibles para procesadores. Esto sucede cuando el usuario adquiere una Processor License (Licencia de Procesador) por cada procesador en el servidor en el

cual el software esté ejecutándose. Una Licencia de Procesador incluye acceso ilimitado de usuarios para conectarse tanto por una red local (LAN) o red amplia (WAN) o por fuera del firewall. No es necesario adquirir licencias adicionales de servidor, CAL, o licencias de Internet Conector. Actualmente este modo de licenciamiento está disponible para sistemas como Windows NT Server.

- ***Licenciamiento por "correo" (Solo para entidades educativas).***

Si una entidad educativa se encuentra usando Exchange Server, tiene la opción adicional del Licenciamiento Por Correo. Con este modelo, se adquiere una CAL para cada correo que acceda al Exchange Server. Si se escoge este modo, después no podrá cambiarse al modo Por Usuario/Por Equipo. Sin embargo, si se elige el modo Por Usuario/Por Equipo después puede cambiar al modo Por Correo, pero sólo una vez. Esta decisión es permanente.

1.4.3 Licenciamiento por Volumen.

Las licencias por Volumen son una forma flexible y económica de adquirir desde cinco hasta miles de licencias. El uso de estas licencias ofrece importantes ahorros, procesos de adquisición flexibles, numerosas opciones de pago y otros beneficios.

También permite obtener a la empresa el derecho a utilizar la tecnología del mañana a precios de hoy y permite además la posibilidad de fraccionar los pagos de las licencias de software Microsoft en anualidades, lo cual reduce el impacto en su flujo de efectivo al realizar adquisiciones de licencias de software Microsoft. Las organizaciones que poseen licencias cubiertas con Software Assurance poseen el derecho a utilizar las últimas versiones disponibles en el mercado de los productos que posean bajo dicha opción durante el período de cobertura del contrato.

El uso de este producto trae grandes beneficios. Entre ellos se pueden mencionar:

- **Mejor capacidad al presupuestar mediante pagos anuales**. Al fraccionar los pagos en anualidades, se puede fijar el precio de las licencias y planificar los presupuestos de adquisición de licencias de software Microsoft con hasta 3 años de antelación.

- **Administración e Implementación de software más eficiente**. Los recursos de formación y las herramientas incluidas con Software Assurance facilitan la labor a los profesionales para implementar

y administrar nuevas versiones de software, actualizaciones y otros contenidos en los PCs y servidores de la entidad.

- **Mejoras en productividad a través de recursos de formación y soporte**. Software Assurance no solo ofrece a los usuarios la posibilidad de utilizar el software más reciente, sino que ofrece además recursos de formación y soporte para que puedan beneficiarse de la tecnología más innovadora.

- **Derechos de Uso en el Hogar**. Con Software Assurance, los empleados podrán trabajar desde su hogar obteniendo una licencia de Microsoft Office para instalar y utilizar en el PC de su hogar.

1.4.4 Llaves de Hardware.

Este método ofrece a los integradores de software la capacidad de distribuir licencias para las aplicaciones, controlando y protegiendo los datos desde una única llave USB lo cual es beneficioso ya que hace crecer el negocio a través de la protección del software y de la propiedad intelectual, así como de la distribución segura de licencias.

Combinar la protección basada en hardware y software en una única solución integrada permite a los vendedores de software la flexibilidad de elegir qué nivel de protección se ajusta mejor al presupuesto y a las necesidades empresariales.

Una llave es un sistema único de distribución sencillo y completo que proporciona a los desarrolladores de software la capacidad de distribuir fácilmente el producto completamente protegido y sujeto en su totalidad a licencias, de manera que se reducen los pasos de instalación para los usuarios finales.

Este método puede utilizar software de partición para ofrecer funciones de almacenamiento masivo a los usuarios finales. Esto garantiza una notable flexibilidad, puesto que permite a los integradores de software utilizar la llave del modo que deseen, por ejemplo, designar un espacio para el software (ROM) y otro para el almacenamiento de datos para el usuario final, de manera que éste pueda almacenar archivos de datos de aplicación propios. Además, se puede ejecutar la aplicación protegida directamente desde la llave, lo que redunda en una mayor comodidad.

Entre los ejemplos de sistemas que actualmente utilizan este método están: USB HASP HL Drive de Aladdin Knowledge Systems, Llaves de Hardware Sentinel, Llave USB Virtual Unikey de SecuTech, entre otras.

1.4. Sistemas automatizados existentes.

Protector de Licencias (License Protector) 2.5

License Protector es un sistema que administra licencias y módulos. Además, puede generar versiones de demostración y de tiempo limitado (versiones de prueba). Proporciona un programa de protección contra copia y soporta pruebas de usuario concurrentes. (4)

Genera automáticamente licencias y se activa a través de Internet. No incurre en pagos por derechos de uso. Está disponible en 10 idiomas. Tiene un costo de $ 375,00 USD y su tamaño es de 14846 K. Fue diseñado para las plataformas de Windows.

Al igual que otros sistemas reúne un conjunto de cualidades que lo hacen efectivo en el proceso de licenciamiento, pero su adquisición es muy costosa por lo que muchas entidades se ven privadas de su uso.

CryptoLicensing. Net 2010

CryptoLicensing es una solución 100% para .NET que tiene como funcionalidades agregar licencias, protección contra copia, activación de hardware, capacidades de bloqueo a la red, Windows Forms (WinForms), aplicaciones de WPF, los componentes, controles y los sitios web ASP.Net.

CryptoLicensing usa la fuerza militar más reciente, el estado de la más avanzada tecnología de cifrado para generar licencias de seguro irrompible y con ello asegurar la protección del software y la propiedad intelectual.

Es un sistema propietario cuyo costo por utilización se eleva a los $ 149 USD solo en su versión estándar para una sola licencia de desarrollo. Tiene características que permiten al usuario proteger de forma efectiva las soluciones que realiza, sin embargo, como su principal desventaja indica, es difícil obtenerlo ya que no todas las entidades tienen el capital suficiente para respaldar los gastos por instalación que se generan. En esta situación se encuentra el Departamento de Software Médico Imagenológico del CESIM.

DeployLX licencias V3.2

DeployLX permite la concesión de licencias y ha sido el principal instrumento de protección de copia para .NET por más de 6 años. Garantiza la seguridad sin precedentes y la flexibilidad que pueden proporcionar fácilmente los ensayos, el hardware de bloqueo y control remoto para los productos.

Este administrador de licencias hace que sean fácil de diseñar las reglas de licencia basada en el uso de cualquiera de los 24 límites incluidos. La interfaz de usuario causa una gran primera impresión y hace que sea fácil para los usuarios entender y trabajar dentro de los términos de licencia que se especifique.

Es un sistema de licenciamiento efectivo, del tipo propietario por lo que esa viene a ser una de sus principales desventajas. Además, aunque permite la posibilidad de descargarlo desde un ordenador personal este proceso se limita solo a algunos países desarrollados como son: Estados Unidos, Canadá, Francia, Alemania, entre otros.

Protection Plus V4.6.0.1

Protection Plus es un poderoso conjunto de herramientas flexibles para la concesión de licencias de software que permite proteger las aplicaciones de forma remota, mediante la gestión del acceso a los clientes. Proporciona todo lo necesario para garantizar los canales de distribución ya existentes y crear otros nuevos.

Es usado por miles de desarrolladores de todo el mundo desde hace 10 años. PLUS es una tecnología probada en el campo. Posee características de prevención de la piratería opcional que pueden identificar los ordenadores, discos duros, y servidores de red. Los códigos de activación son únicos y no pueden volver a utilizarse.

Su costo se eleva a los $342USD por cada PC donde vaya a instalarse, lo que provoca que su utilización se vea afectada por parte de muchos desarrolladores de software. El CESIM no cuenta con los fondos suficientes para suministrarle este producto al Departamento de Software Médico Imagenológico por lo que es imposible su empleo en la protección y el licenciamiento de las soluciones que se desarrollan en él.

Softshell Software LicensingSystem 1.0.0

Es un sistema de bajo costo, muy flexible y eficaz en la adquisición de licencias de software. Requiere de muy poca programación. El usuario puede crear un software completamente funcional en el tiempo que se especifique. Softshell no requiere que se bloquee la aplicación de hardware. Opcionalmente, puede

bloquear el software a una cadena personalizada, como el nombre, dirección de correo electrónico o nombre de la empresa.

Otras características incluyen: bloqueo a múltiples tipos de hardware (disco duro, número de serie, dirección MAC, etc.). Tiene hasta 24 opciones de licencia y permite que el registro caduque después de un período de tiempo especificado. De hecho, Softshell permitirá que la aplicación se ejecute en modo invitado. Funciona en la mayoría de los sistemas operativos de Microsoft Windows NT - incluyendo Windows 7, en la mayoría de las máquinas virtuales, y trabaja en las de 32 bits y 64 bits.

Entre sus principales características esta que cada aplicación a desarrollar está protegida por una contraseña única. Nadie más - incluyendo los desarrolladores de Softshell - tendrá acceso a la información de registro. No se requieren derechos administrativos - una vez que los archivos básicos se han instalado, la aplicación no requiere derechos de administrador para ejecutarse.

De hecho, la aplicación estará totalmente protegida durante el período de evaluación mientras se ejecuta bajo una cuenta de invitado. Se ejecuta en los sistemas operativos Windows 2000 y Windows 7 en el de 32-bit (x86) y 64-bit (x64) a modos de transporte y en Máquinas Virtuales compatibles (VMWare y Virtual PC) al igual que cualquier otro equipo.

Este sistema tiene características que pueden ser las necesarias para suplir las necesidades del departamento. Es de fácil uso y su adquisición es gratuita. Puede considerarse como un candidato a utilizar en el proceso de licenciamiento para los software que se desarrollan en el Departamento de Software Médico Imagenológico.

SerialShield SDK.

Es un sistema de protección anti-copia y de licencia para desarrolladores profesionales que quieren tener un control total de su software, sin costes de autor. Serial Shield API se puede usar con Microsoft Visual Basic, VBA Access, VBA Word, VBA Excel, Microsoft Visual C++ o Borland Delphi y C++ Builder y otros lenguajes.

Se utiliza además para dar capacidad de evaluación a su aplicación y maneja todos los aspectos para asegurar que su aplicación esté dentro del período de evaluación y registrado perfectamente en la máquina específica. (5)

Utiliza un código especial para vencer a la mayoría de los debuggers y previene contra intentos de espionaje del código, incluyendo W32Dasm, Softice, TRW 2000, InCTRL5, Turbo Debugger, Sourcer, Filemon, ExeSpy, ResSpy, RegMon y Memory Monitor y todas las que son herramientas comunes en el mundo de los crackers. Puede proteger aplicaciones desarrolladas y que funcionan bajo Win95, Win98, WinNT y Win2000.

Tiene algoritmos de protección de software diferentes de otros esquemas de protección, porque soporta todas las técnicas y características de seguridad de datos de última generación:

- Protección anti-debugging, anti-rastreo y anti-monitorización.
- Tecnología de codificación segura AES Rijndael para proteger el Serial Key.
- Cierre de licencias Machine ID para prevenirse contra instalación en una máquina ilegal.
- Detecta si se cambia la fecha o se reinstala el demo para ganar un uso adicional.
- Generador de Serial ID para identificar cada copia y cerrar la clave con este Serial ID.
- El Servidor de Internet Serial Shield permite al usuario final obtener una clave de activación completa a través del correo electrónico.
- Protección de software para Visual Basic, VBA, Delphi, C/C++, otros.
- Creación de copias de evaluación con su software por días o por usos.
- Alquile/arriende el software a sus clientes.
- Envía a sus clientes una demostración de su software antes de comprarla para que la evalúen.
- Distribuye su software vía internet, CD-ROM o diskette.

Este sistema fue comprado por parte de la dirección de producción de la Universidad para utilizarlo en el licenciamiento del producto alas PACS-alas RIS. En esta investigación se analizará la posibilidad de emplearlo como herramienta para el desarrollo y aplicación de la estrategia de protección que se propondrá.

1.5. Tendencias actuales del licenciamiento.

Hasta hace poco, casi todas las licencias eran compradas sobre una base de "licencias perpetuas", o sea, los compradores pagaban una vez y usaban el software por tanto tiempo como ellos quisieran. La tendencia es lograr relaciones de largo plazo con los usuarios, más que vender "licenciamiento perpetuo" que luego hay que actualizar. Las compañías de software lanzan "versiones mayores" una vez al año o cada dos años,

sin embargo, durante el año se liberan una serie de "versiones menores" que reparan "bugs", dan mayor estabilidad al software o entregan nuevas capacidades.

Entre las tendencias más reconocidas actualmente se encuentra el ***Licenciamiento Perpetuo*** que se define como un esquema de licenciamiento por volumen fácil, flexible y diseñado para reducir los costos de adquisición, actualización, mantenimiento y administración de software en todas las computadoras y servidores. (6)

Hoy en día se conocen dos tipos fundamentales de licenciamiento perpetuo:

Open académico que es un programa de licenciamiento por volumen para instituciones educativas que requieren un mínimo de 5 licencias.

Select académico que es un programa de licenciamiento por volumen para instituciones educativas con más de 250 computadoras.

Los dos pueden incluir sólo la licencia o la licencia+ Software Assurance.

Los ***Contratos de Mantención o Actualización*** también conocidos como CMS Superior conforman otra de las tendencias del licenciamiento. Tienen la ventaja de permitir al usuario estar siempre actualizado con las últimas versiones. Pueden ser de un año, dos años o más. Para las compañías de software estos contratos les permiten crear una relación más cercana y de largo plazo con las empresas, así como un flujo esperado de ingresos. El valor de un Contrato de Actualización es aproximadamente el 25-30% del valor de la licencia. (7)

Estos contratos tienen una serie de ventajas que posibilitan una mayor comodidad para el usuario. Elimina los costos inciertos ya que la mantención del sistema adquirido no depende de la cantidad y complejidad de cambios normativos, ahora el precio es fijo independiente de los cambios que se produzcan en el año, incluyendo además todas las mejoras incorporadas al sistema durante la vigencia de su contrato de mantención.

Permiten además una mejoría en la calidad del soporte garantizando que el usuario posea la última versión del sistema, asegurando con esto mayor calidad en sus operaciones. Todo esto posibilita que se alcance un mayor grado de madurez en la aplicación y permite darle la mejor atención y servicio a la misma así los técnicos conocen cuál es la versión en uso. Esto implica que sea económicamente más conveniente ya que

la suma que se desembolsa en el contrato es mucho menor que la que el usuario debía pagar antes cuando no existía este método.

Mientras existe cierto valor de ser propietario de por vida, hay beneficios mayores al "arrendar". La nueva modalidad es el ***Arrendamiento de Software o Suscripción***, el cual permite usar el software por un período limitado, sin tener que incurrir en el costo de comprar licencias (costo de propiedad). Por otro lado, el vendedor evita los largos períodos de evaluación y logra establecer una relación comercial a largo plazo.

Se basa en un modelo de distribución en el cual el software está hospedado en un servidor y al que los usuarios acceden a través de una conexión de red, lo que les permite a las empresas no tener que mantener su software ni costear su soporte. Con una suscripción o mediante el pago por su uso las compañías de pequeña dimensión pueden contar con las mismas herramientas que las de mayor volumen, elevando su competitividad a través de la reducción de costos.

Sin embargo, y pese a todos los beneficios que ofrece, en el alquiler de Software aún persisten reticencias por parte de algunos sectores empresariales ante las preocupaciones existentes en torno a temas de seguridad, pérdida de control y fiabilidad. Parten de la base de que si el alquiler de Software se estandariza va a resultar difícil que los proveedores puedan mantener la integridad y confidencialidad de sus clientes.

La personalización del consumo, donde el usuario final en cada área de negocio decide por el uso de determinadas aplicaciones cada vez más repartidas, facilita la adopción de servicios muy precisos y la medida operativa para empresas de diferentes tamaños. Esta tendencia es la que impulsa que sean cada vez más las compañías que buscan el software ya no como una herramienta pasiva sino proactiva frente a las coyunturas operativas del negocio

Software-as-a-Service **(SaaS)** empuja a los proveedores de licencias tradicionales a cambiar al modelo de pago por uso, que típicamente viene con un paquete integrado de licencia, mantenimiento y actualización, proporcionando a los clientes una sensación de licenciamiento más dinámica y sencilla.

El SaaS es un modelo de distribuir aplicaciones de computación por medio de Internet. Los usuarios de las aplicaciones de software SaaS no pagan licencias para instalarlo en sus computadoras. En lugar de ello pagan una suma mensual por usarlo. El término SaaS se ha convertido en el preferido de la industria, reemplazando a los que se han estado utilizando como “On-Demand” o ”Utility Computing”.

Se basa en que los datos y programas sean almacenados en un ambiente seguro centralizado, que sea de fácil acceso y sencilla administración. Cada usuario en la red tendrá su propio perfil, accesible desde un directorio común, sin estar atado a una computadora específica. Los usuarios almacenan sus datos en un repositorio central y no en máquinas locales. Las aplicaciones y servicios son manejadas desde ese directorio común, con accesos predefinidos de acuerdo con los roles de los usuarios, en su grupo correspondiente. (8)

Diferencias críticas entre el modelo de SaaS y el modelo tradicional de licencia están conduciendo a la adopción de SaaS. En el modelo tradicional el cliente adquiere una licencia perpetua y asume la responsabilidad de manejar el software. En este modelo hay un alto costo inicial asociado a la compra de la licencia, así como la responsabilidad de la puesta en marcha y del mantenimiento correspondiente. Debido a los rápidos cambios tecnológicos, las costosas aplicaciones de software se convierten rápidamente en obsoletas.

Es por eso que no importa cómo se le termine llamando, SaaS es sin duda el futuro de la industria del software. Las ventajas de uso inmediato, pago por uso, actualizaciones inmediatas, recuperación instantánea de fallas y no tener que preocuparse del mantenimiento han capturado de inmediato al mercado.

Empresas como Microsoft han revelado nuevas estrategia para enfrentar este tipo de competencia. Específicamente la entidad antes mencionada desarrolló un programa Web llamado "MS CRM Live", el cual ha estado distribuyendo por medio de asociados, con tecnología cliente-servidor, y ocasionalmente instalándolo en servidores de sus asociados.

El hecho de que Microsoft comience a ofrecer servicios en demanda, ratifica la nueva tendencia mundial de distribución de programas de computación basados en la Web. Los usuarios de MS CRM Live podrán accederlo usando cualquier navegador o por medio de Microsoft Outlook. El servicio vendrá en dos sabores, edición Profesional y edición Empresarial, con características y precios diferentes.

Otra de las tendencias son los ***Modelos de licencia híbridos*** que han surgido con el objetivo de cubrir mejor las necesidades de los clientes y poder cumplir objetivos estratégicos de negocio. Las empresas piensan que adquieren una ventaja estratégica por disponer de una escala variable de términos de licencia

para los distintos productos que componen su catálogo. Esta ha sido la forma dominante de las licencias de software en el sector. Los siguientes ejemplos dan una muestra de esta tendencia:

Apple Computer ha combinado el software comunitario con el software comercial para crear el sistema operativo OS/X. Aprovechando un derivado de la versión Unix BSD de 32 bits gratuita, la empresa ha podido migrar en poco tiempo hacia un sistema operativo de 32 bits sin tener que reescribir todo su sistema operativo anterior de 16 bits.

En el caso de Real Networks ha publicado boques de código fuente de su reproductor multimedia bajo los auspicios del proyecto Helix. El software fue originalmente desarrollado como producto empaquetado tradicional sin publicación del código fuente. Al abrirlo a la comunidad de desarrollo, Real Networks consiguió generar mucha más actividad de desarrollo alrededor de su tecnología, y con ello mayor oportunidad de negocio para la propia empresa.

Microsoft comparte código fuente con clientes, asociados, gobiernos y desarrolladores en todo el mundo a través de su Iniciativa de Código Compartido. El 90 por ciento de las más de 80 ediciones de código fuente compartido permite a los desarrolladores modificar y redistribuir el código fuente.

Esta ofertas incluyen tecnologías que van desde sistemas operativos enteros, como Windows XP a herramientas de desarrollo como el Windows Installer XML, o aplicaciones como FlexWiki. Microsoft seguirá ampliando sus ofertas de Código Compartido como una manera de permitir el acceso a los desarrolladores a la tecnología que necesitan para crear nuevos y potentes productos de software.

Estudios actuales indican que la tendencia de licenciamiento ***Código Abierto (Open Source)*** se ha venido presentando como la alternativa principal. Esto no afirma que pueda desplazar a las grandes soluciones propietarias de las compañías de software a mediano o largo plazo, pero sí creará una saludable presión en los precios de los productos en áreas como las de sistemas operativos, herramientas de desarrollo en tecnologías de bases de datos, entre otros, sectores en que el código abierto se convertirá en una alternativa real y atractiva, y con ello un factor que facilitará la reducción de los costos.

1.6. Tendencias del licenciamiento en Cuba.

Los productos que son desarrollados en la industria del software cubana son mayormente dentro del campo del Software Libre, por lo que los métodos de licenciamiento que se llevan a cabo para proteger dichas

soluciones requieren de las licencias específicas para este tipo de software. Principalmente se utiliza la GPL del proyecto GNU Linux por ser libre para su uso y de gran demanda.

En instituciones como la Universidad de las Ciencias Informáticas (UCI) se desarrollan proyectos destinados a la producción tanto del software libre como del software propietario. Esto se debe a que esta entidad es una de las principales encargadas del desarrollo del software en la industria cubana.

Específicamente en el Departamento de Software Médico Imagenológico del Centro de Informática Médica (CESIM) radicado en dicha universidad se han realizado varias soluciones de software propietario. Debido a las características de las mismas se emplea como método de protección el licenciamiento mediante Llaves de Software empleando datos del hardware.

Este tipo de protección consiste en programar la protección a partir de los datos del hardware del equipo donde va a ser instalada la aplicación, es decir, durante la instalación de la misma se comprobará si la información que fue recogida para la implementación del mecanismo de seguridad coincide, en caso de ser positivo permitirá que sea instalada la solución, en otro caso este proceso finaliza, afectando así en alguna medida la integridad de la instalación del software.

Al utilizar la técnica antes mencionada se genera una dependencia total por parte del cliente hacia el proveedor, ya el despliegue de estas soluciones exige el pago de la licencia por cada estación en la que el software esté siendo utilizado.

Este método provoca además que cada vez que se necesite realizar cualquier modificación o actualización en el software el usuario deba remitirse a la empresa dueña del mismo y hacer la solicitud, ya sea directamente, por correo electrónico u otra vía, de lo contrario se incurre en una violación de las regulaciones del proceso de licenciamiento. Lo anteriormente expuesto garantiza que el costo por utilización de las soluciones sea elevado contribuyendo así al incremento de capital en la economía del país.

Conclusiones.

En este capítulo se ha analizado las características fundamentales del licenciamiento y como se ha transformado. Este ha logrado sostenerse hasta hoy como un modelo basado en desarrollar un programa informático para una vez terminado sea distribuido de forma creciente, haciendo que los gastos de producir

nuevas copias sean mínimos. Por mucho tiempo se ha mantenido en su pedestal ya que la compra y venta de licencias de software se ha realizado bajo los esquemas estructurados por las grandes desarrolladoras.

Si bien los clientes han tenido la posibilidad de elegir entre distintas empresas de software y las licencias de uso pueden ser distintas, su contenido esencial es el mismo y además de estandarizado, es cerrado a los ojos del cliente, sin mucho interés de los productores en discutir las condiciones de las licencias y los límites de propiedad que estas permiten.

El licenciamiento persistirá, esto será principalmente por las ofertas existentes de licenciamiento a perpetuidad, es decir, que una empresa puede contar con el uso de una plataforma y sus actualizaciones para toda la vida con un pago único anual por debajo del costo de las soluciones cerradas.

CAPÍTULO 2. CARACTERÍSTICAS DE LOS MÉTODOS DE LICENCIAMIENTO DE SOFTWARE.

Posterior al estudio de todos los epígrafes vinculados al marco teórico de la investigación que se desarrolló en el primer capítulo de este trabajo de diploma, se puede dar paso entonces al comienzo del segundo capítulo, donde serán abarcados temas como: la realización de una descripción detallada de los métodos de protección y licenciamiento existentes, así como el análisis y la comparación entre los más empleados.

2.1. Protección de software.

Para la industria del software, la protección de sus productos es una característica importante, no sólo en cuanto a las copias ilegales del software, sino también a la protección de los derechos de propiedad intelectual del código.

Se puede concluir que un producto de software es caro de desarrollar, pero barato para reproducir, a causa de su naturaleza digital. La tecnología digital plantea dos propósitos para la gestión de derechos. En primer lugar, reduce el costo de hacer copias, y en segundo lugar, permite que las copias sean distribuidas rápidamente, fácilmente y en forma barata.

Es por eso que para garantizar la seguridad del software se deben cubrir una variedad de técnicas que van desde protección legal por derechos de copia (copyright), licencias y patentes, a métodos técnicos.

2.2. Métodos de protección legales.

2.2.1. Copyright.

Es un derecho exclusivo de un autor para controlar la distribución y reproducción de su trabajo original. En software, tanto el código fuente (legible por personas) como el código objeto (ejecutado por la máquina), y los manuales relacionados, son los que se eligen para proteger por copyright. Pero los métodos y algoritmos dentro de un programa no son protegidos. (9)

La ley de copyright es un mecanismo clave de protección legal, ya que se puede aplicar virtualmente a todo el software de una computadora. La copia del software protegido por copyright sin el permiso de su propietario puede exponer al que lo copia a penalidades criminales. Esto es muy importante para prevenir el robo y piratería de productos de software, mientras que alienta a los desarrolladores a generar investigaciones en nuevos productos y servicios.

Esta técnica tiene como ventajas su bajo costo, facilidad de obtención y velocidad de implementación, pero cabe destacar que aunque la disponibilidad de la protección por copyright se ha incrementado significativamente en el mundo, su cumplimiento sigue siendo un problema práctico principal ya que no evita en su totalidad que el software sea pirateado.

Es por eso que se considera como un método utilizable, pero siempre acompañado de otra estrategia de protección ya sea mediante software o mediante hardware, es decir, acoplado a otro método de protección técnico.

2.2.2. Patentes.

Una patente es un derecho legal provisto por una entidad gubernamental que permite a un inventor prevenir que otros fabriquen, vendan o usen la invención del propietario de la patente. (10)

La protección por patentes puede convertirse en una herramienta competitiva valiosa, comparada a la protección tradicional por copyright. Contrariamente al copyright, una patente protege ideas y algoritmos en un producto de software, en lugar del código en sí mismo. Un ejemplo típico es la protección de funciones, métodos, sistema y algoritmos, así como se aplica a las fórmulas matemáticas.

La razón principal de que su uso no sea muy demandado es que son muy caras de obtener, en general el proceso de aplicación toma varios años. Además, la revelación de la gran cantidad de información del producto que se requiere puede poner en riesgo la confidencialidad de la información.

Por lo tanto, la protección por patentes es más adecuada para software central o de larga vida. No todo el software puede ser patentado; como con cualquier invento, los requerimientos, tales como novedad y no obviedad permanecen.

2.2.3. Licencias.

Gran parte del software de hoy en día no se adquiere por los usuarios, sino que se “licencia” a los usuarios. En este caso, el acuerdo de licencia establecerá qué derechos y privilegios se conceden a quien obtiene la licencia, mientras que otros derechos permanecen con el propietario del copyright.

Un propietario de software típicamente concede un derecho no exclusivo (licencia) a un usuario para usar una copia de su software y prohíbe fomentar la copia y distribución del mismo a otros usuarios.

Un esquema de licencias formulado claramente y consistente siempre será benéfico al tomar acciones contra el copiador ilegal. Establece el límite entre los actos legales e ilegales por el que obtiene la licencia con respecto a su trabajo.

2.3. Métodos de protección técnicos.

Primeramente se debe entender algo, la garantía legal de los derechos exclusivos de protección de software por medio de patentes, derechos de copia (copyright) y licencias no proveen un control completo sobre el uso del producto original. Por ende surge la necesidad de tener que reforzarlos por medio de mecanismos técnicos. Estas técnicas incluyen programas y dispositivos capaces de prevenir el uso no autorizado de software.

Los métodos técnicos se basan en dos aproximaciones:

• Protección mediante software.

• Protección mediante hardware.

Los métodos de protección basados en hardware proveen principalmente medidas preventivas, mientras que los métodos basados en software proveen tanto medidas preventivas como medidas para detectar errores.

2.4. Métodos de protección basados en Software.

2.4.1. Protección por número de serie o palabra clave.

La protección por número de serie es uno de los métodos más antiguos para restringir la distribución de copias ilegales de software. Consiste en ingresar, cuando se instala el software, una palabra clave conformada por números y caracteres. Adicionalmente, el software puede requerir el ingreso de otros números de serie para acceso a diferentes funciones del programa.

Los dos métodos más comunes para la utilización de números de serie son:

• En la instalación del software: antes de comenzar cualquier tarea relacionada con la instalación (copiado de archivos, petición de datos, etc.) se solicita al usuario el número de serie para verificar que él posee una copia original del software; ese número de serie estará generalmente impreso en el CD o en el manual de operaciones.

• Cada vez que se inicia el software: este método es el más usado por juegos interactivos, que requieren el ingreso de un código de acceso en base a datos que se proveen por pantalla. Los códigos de acceso están impresos en un manual, con todas las posibles combinaciones de datos.

Una variación de este método se está utilizando actualmente por grandes empresas como Microsoft. Consiste en la validación de los números de serie a través de Internet: cuando el usuario ingresa el número de serie, la aplicación se conecta con un servidor remoto seguro para validar el número; si éste no existe en su base de datos, el programa de instalación finaliza diciendo que el número no está autorizado. Este método se utiliza por el sistema operativo Windows XP.

Hablando de ventajas de este método se puede citar que **i**mpide la distribución de copias ilegales de software. Y como desventaja existe la posibilidad de que si se descubre el código de validación de un número de serie se pueden crear aplicaciones generadoras de números de serie para el producto.

2.4.2. Protección por tiempo.

Pueden operar de distintas formas:

1- El software comprueba si han transcurrido X días desde su instalación, y si es así procede a su salida inmediata o en el peor de los casos a su desinstalación automática. Durante la salida/desinstalación del software, éste puede mostrar algún mensaje informando al usuario del hecho en cuestión. Puede pasar que el software vuelva a funcionar durante X días si se vuelve a instalar. (11)

2- El software comprueba si ha llegado a una fecha límite; si es así, procede de la misma manera que en el caso anterior. La diferencia está en que el software dejará de funcionar a partir de una fecha determinada y no funcionará si se vuelve a instalar.

Existen algunas variantes específicas para este tipo de protección, entre ellas se pueden encontrar:

- **Por fecha** (obtención).
  La forma más difundida de protección es la que le permite al futuro comprador evaluar el programa un número de días preestablecido. La fecha de inicio se obtiene el día que se instala el programa, y algunas veces se guarda el día que expira la demo.

Hay diversas maneras de almacenar esta fecha en la PC. Una es grabarla como una entrada en el registry (base de datos que siempre tiene el Sistema Operativo (OS) disponible), y la otra es con un archivo en alguno de los discos rígidos que tenga la PC.

- **Por Fecha** (verificación simple).
  Las primeras protecciones de este tipo verificaban solamente que la fecha actual fuera menor que la de expiración. De este modo, cuando se instala el programa, con solo adelantar la fecha en varios años se obtiene una fecha de expiración muy lejana a la fecha actual.

- **Por Fecha** (verificación del límite superior e inferior).
  Para superar la anterior debilidad se guarda la fecha de instalación. Primero se verifica que la fecha actual sea menor que la de instalación, en caso afirmativo se da por finalizado el período de demo.

  Después se le suman a la fecha de instalación los días que estableció el fabricante, y se la compara con la fecha actual: si es menor se da por terminado el período de demo. También se puede quebrar la protección anotando la fecha en que se hizo la instalación, para cambiarla todas las veces que se corre el programa.

- **Por veces de uso.**
  Una manera de evitar el quiebre de la protección al cambiar la fecha es tener un contador de veces de funcionamiento: este se incrementa cada vez que se corre el programa. El inconveniente de este método es que si se corre la aplicación y no se cierra, los días transcurren y el contador no se incrementa.

- **Por veces de uso** (contador de 24 horas).
  Para evitar que al dejar funcionando la PC indefinidamente se anule la protección, los fabricantes de software decidieron agregarle un contador de días transcurridos mientras el programa está en funcionamiento; por cada día que transcurre se incrementa el contador de veces de ejecución. Cuando llega a la cantidad de veces establecida la demo expira.

Este método es usado principalmente por las empresas fabricantes de software para distribuir sus programas por Internet. Si el interesado quiere comprar el programa, el fabricante le envía un código de activación por e-mail para remover la limitación temporizada.

El principal punto débil de las versiones trial o demo que se realicen de las aplicaciones que utilizan este método es que hay que dejar una marca en la PC para indicar en qué fase está: demostración, expiración, o versión full operativa. Es por ello que se deben emplear herramientas diseñadas para autenticar la actividad de un programa con respecto al acceso a los discos. Entre ellas se pueden analizar:

- **Reloj en tiempo real** (Fecha actual).

  Una manera de comenzar con el estudio de una protección por tiempo es encontrar en qué punto del programa se necesita saber la fecha actual. El sistema operativo entre otras cosas, controla el comportamiento del reloj de la máquina (RTC); para esto provee funciones en el formato de APIs que, por ejemplo, permiten saber la fecha y hora actual.

  La gente que hace ingeniería inversa conoce esta característica, sabe que sin importar el lenguaje de programación en que se escribió el programa, cuando el mismo requiere la fecha actual para determinar si el programa expiró o no, tiene que llamar a una función conocida del OS.

  Luego corre el programa con un debugger[1] y pone un BPX[2] en esta API. Cuando se detiene la ejecución se está en las inmediaciones de la rutina de verificación de expiración.

- **Búsqueda de la protección por MsbBox.**

  Otra técnica muy difundida en el ambiente de la ingeniería inversa es usar el mensaje de demo expirada (NAG), para detener la ejecución del programa y llegar al punto donde se decidió que la fecha o veces de ejecución habían expirado. Cuando se usa esta técnica y el BPX pica (se detiene la ejecución del programa), significa que se está en las inmediaciones de la rutina de verificación.

- **Clave de activación.**

  Como este tipo de protección tiene que ingresar una contraseña cuando se hace la compra de la versión, se pueden usar las mismas técnicas de remoción que se usan para las protecciones de software por clave.

---

[1] Software para ejecutar paso a paso una aplicación.
[2] Experto en proceso de negocios.

Las principales desventajas con que cuenta este método son: que el software debe saber cuándo se instaló por primera vez, guardando esta información en algún lugar, probablemente en algún archivo propio o en el registro del sistema operativo y debe comprobar la fecha en el momento en el que el programa está siendo ejecutado, y hacer los cálculos correspondientes.

Además, si el ejecutable principal realiza los controles en una librería externa (DLL) esto hace al software más inseguro ya que si se elimina la protección en la librería, se desprotegen todos los programas que dependan de dicha librería.

2.4.3. Ofuscación.

Intuitivamente la ofuscación del código tiene como propósito realizar modificaciones en el código compilado con la intención de hacerlo ilegible. La ofuscación del código puede ser más difícil de reconstruir si además de lo anterior se agrega el paso de variables entre los procedimientos, la idea es utilizar un arreglo global para pasar los valores asignados por el despachador de variables, así cada vez que se llame a la función se pasará el valor al arreglo de variables global, logrando de esta manera que los valores asignados cada vez que se llama la función no sean constantes, sino que al examinar el código ofuscado no sea evidente el proceso que se realizó en la llamada a la función. (12)

La mayoría de las técnicas existentes son teóricas, solo existen en documentos científicos. Sólo un pequeño subconjunto ha sido implementado en productos comerciales o aún en prototipos.

La ofuscación de código dinámica es una nueva técnica que tiene varias ventajas y desventajas, dependiendo del punto de vista y uso que se le dé a la misma. Pues si es aplicada como un mecanismo de seguridad, puede ser muy prometedora, ya que aunque sea una técnica que puede ser vulnerada mediante el uso de algunos análisis estáticos y dinámicos, proporciona un nivel un poco más elevado de seguridad.

En cambio, si es utilizada como un arma para lograr burlar los actuales mecanismos de defensa puede ser muy peligrosa, pues aunque se podría detectar su peligrosidad mediante algunos de los análisis mencionados, consumiría mucho tiempo y lo más probable es que mientras se realice este tipo de análisis, se descuide algún otro aspecto de la seguridad.

Las técnicas de ofuscación se pueden dividir en cuatro categorías:

1. Ofuscación de diseño (Layout Obfuscation).

Consiste en cambiar o remover información del código ensamblador que no modifica parte del código ejecutable. En general la información de debugging y comentarios que permanecen dentro del código y son los datos útiles para la ingeniería inversa.

2. Ofuscación de dato (Data Obfuscation).

Esta clase de ofuscación no modifica el código, se basa en ofuscar los datos y las estructuras de datos embebidas en los programas.

3. Ofuscación de control (Control Obfuscation).

El objetivo aquí es alterar el código, más precisamente alterar el flujo de control en el código y no la parte computable del mismo. Algunas formas de hacer estas modificaciones surgen directamente de las optimizaciones del compilador y de la reingeniería de software.

4. Transformación preventiva (Preventive transformation).

Aquí el punto es claramente el descompilador/desofuscador y no la ingeniería inversa. Hay dos tipos de transformaciones posibles: hacia un blanco, que explora las debilidades en los descompiladores y desofuscadores actuales, y las inherentes que averiguan los problemas inherentes con técnicas de desofuscación conocidas.

Esta técnica es muy utilizada en la actualidad en sistemas como el Java Scripts Obfuscator v1.0,SG*Software*Gurú,SecureEngine® de OreansTechology, en sistemas desarrollados por los laboratorios Kasperky (antivirus),Freeware .NET ObfuscatorSkater Light 1.03.0 de RustemSoft, entre otros.

2.4.4. Encriptación de datos.

En un Sistema de Comunicación de Datos, es de vital importancia asegurar que la información viaje segura, manteniendo su autenticidad, integridad, confidencialidad y el no repudio de la misma, entre otros aspectos. Estas características sólo se pueden asegurar utilizando la Encriptación de Datos y las Técnicas de Firma Digital Encriptada.

Métodos de encriptación:

Para poder encriptar un dato, se pueden utilizar tres procesos matemáticos diferentes:

Los algoritmos HASH, los simétricos y los asimétricos.

· **Algoritmo HASH:**

Este algoritmo efectúa un cálculo matemático sobre los datos que constituyen el documento y da como resultado un número único llamado MAC (Message Authentication Code). Un mismo documento dará siempre un mismo MAC.

· **Algoritmos Simétricos:**

Utilizan una clave con la cual se encripta y desencripta el documento. Todo documento encriptado con una clave, deberá desencriptarse, en el proceso inverso, con la misma clave. Es importante destacar que la clave debería viajar con los datos, lo que hace arriesgada la operación, imposible de utilizar en ambientes donde interactúan varios interlocutores.

· **Algoritmos Asimétricos:**

Requieren dos Claves, una Privada (única y personal, solo conocida por su dueño) y la otra llamada Pública, ambas relacionadas por una fórmula matemática compleja imposible de reproducir.

El usuario, ingresando su PIN genera la clave Pública y Privada necesarias. La clave Pública podrá ser distribuida sin ningún inconveniente entre todos los interlocutores. La Privada deberá ser celosamente guardada. Cuando se requiera verificar la autenticidad de un documento enviado por una persona se utiliza la Clave Pública porque ya esa persona utilizó la Clave Privada.

- **Encriptación Asimétrica + Simétrica**

Debido a que la encriptación asimétrica es más lenta que la simétrica, cuando la información a encriptar es mucha, se utiliza una combinación de algoritmos. El algoritmo simétrico se utiliza para encriptar la información y el asimétrico para encriptar la llave del algoritmo simétrico con que se encriptó la información. Entonces, el proceso es mucho más rápido. En cada ida y vuelta al servidor se generan nuevas llaves y se realiza todo el proceso. Windows utiliza esta combinación en la encriptación de los archivos.

Otro motivo para utilizar esta encriptación combinada es la necesidad de encriptar textos largos. La encriptación asimétrica además de ser ineficiente en tiempo, tiene limitaciones de tamaño. El tamaño máximo depende del largo de la llave.

Si se genera una llave, la cual se almacenará en una clase o un servicio, y luego se escoge el método asimétrico que se va a combinar, el proceso puede suceder de la siguiente forma:

1. Se encripta la información con la llave definida mediante un método de encriptación simétrico.
2. Luego se encriptará esa llave mediante el uso de un método de encriptación asimétrica garantizando así que la misma, aunque sea del cocimiento de la clase o servicio antes mencionado, no tenga ninguna utilidad a la hora de desencriptar la información que se desea proteger.
3. La encriptación se produce en un lado y la desencriptación en otro.

2.4.5. Firma Digital.

Es la transformación de un mensaje utilizando un sistema de cifrado asimétrico de manera que la persona que posee el mensaje original y la clave pública del firmante, pueda establecer de forma segura, que dicha transformación se efectuó utilizando la clave privada correspondiente a la pública del firmante, y si el mensaje es el original o fue alterado desde su concepción.

Se intenta hacer coincidir el modelo de firma digital con los requerimientos y virtudes que debe tener una firma y así darle validez a esta mecánica. El objetivo final es el mismo que el de la firma ológrafa: dar asentimiento y compromiso con el documento firmado.

Aspectos técnicos.

A diferencia de la firma manuscrita, que es un trazo sobre un papel, la firma digital consiste en el agregado de un apéndice al texto original, siendo este apéndice, en definitiva, la firma digital; al conjunto formado por el documento original más la firma digital se le denominará mensaje.

Este apéndice o firma digital es el resultado de un cálculo que se realiza sobre la cadena binaria del texto original. En este cálculo están involucrados el documento mismo y una clave privada (que, generalmente, pertenece al sistema de clave pública−privada o sistema asimétrico) la cual es conocida sólo por el emisor o autor del mensaje, lo que da como resultado que para cada mensaje se obtenga una firma distinta, es decir, a diferencia de la firma tradicional, la firma digital cambia cada vez con cada mensaje, porque la cadena binaria de cada documento será distinta de acuerdo con su contenido.

A través de este sistema se pueden garantizar completamente las siguientes propiedades dela firma tradicional:

· Quien firma reconoce el contenido del documento, que no puede modificarse con posterioridad (integridad).

· Quien lo recibe verifica con certeza que el documento procede del firmante. No es posible modificar la firma (autenticidad).

· El documento firmado tiene fuerza legal. Nadie puede desconocer haber firmado un documento ante la evidencia de la firma (no repudio).

Este sistema utiliza dos claves diferentes: una para cifrar y otra para descifrar. Una es la clave pública, que efectivamente se publica y puede ser conocida por cualquier persona; otra, denominada clave privada, se mantiene en absoluto secreto ya que no existe motivo para que nadie más que el autor necesite conocerla y aquí es donde reside la seguridad del sistema.

Ambas claves son generadas al mismo tiempo con un algoritmo matemático y guardan una relación tal entre ellas que algo que es encriptado con la privada, solo puede ser desencriptado por la clave pública.

Resumiendo, la clave privada es imprescindible para descifrar criptogramas y para firmar digitalmente, mientras que la clave pública debe usarse para encriptar mensajes dirigidos al propietario de la clave privada y para verificar su firma.

Si bien no se trata de un tema estrictamente técnico, es conveniente aclarar que en tiempo de generación de cada par de claves, pública y privada, podría intervenir otra clave que es la de la Autoridad Certificante que provee la garantía de autenticidad del par de claves generadas, así como también, su pertenencia a la persona cuya propiedad se atribuye.

Este esquema se utiliza en intercambios entre entidades cuando se trata de transferencias electrónicas de dinero, órdenes de pago, etc. donde es indispensable que las transacciones cumplan con los requisitos de seguridad enunciados anteriormente (integridad, autenticidad y no repudio del origen).

Ventajas Ofrecidas por la Firma Digital.

El uso de la firma digital satisface los siguientes aspectos de seguridad:

· Integridad de la información: la integridad del documento es una protección contra la modificación de los datos en forma intencional o accidental. El emisor protege el documento, incorporándole a este un valor de control de integridad, que corresponde a un valor único, calculado a partir del contenido del mensaje al momento de su creación.

El receptor deberá efectuar el mismo cálculo sobre el documento recibido y comparar el valor calculado con el enviado por el emisor. De coincidir, se concluye que el documento no ha sido modificado durante la transferencia.

· Autenticidad del origen del mensaje*:* este aspecto de seguridad protege al receptor del documento, garantizándole que dicho mensaje ha sido generado por la parte identificada en el documento como emisor del mismo, no pudiendo alguna otra entidad suplantar a un usuario del sistema. Esto se logra mediante la inclusión en el documento transmitido de un valor de autenticación (MAC). El valor depende tanto del contenido del documento como de la clave secreta en poder del emisor.

· No repudio del origen*:* el no repudio de origen protege al receptor del documento de la negación del emisor de haberlo enviado. Este aspecto de seguridad es más fuerte que los anteriores ya que el emisor no puede negar bajo ninguna circunstancia que ha generado dicho mensaje, transformándose en un medio de prueba inequívoco respecto de la responsabilidad del usuario del sistema.

2.4.6. Sistemas de protección/defensa mediante comprobación de la integridad de los datos (CRC).

El CRC es un código de detección de error cuyo cálculo es una larga división de computación en el que se descarta el cociente y el resto se convierte en el resultado, con la importante diferencia de que la aritmética que se utiliza conforma que el cálculo utilizado es el arrastre de un campo finito, en este caso los bits. (13)

El tamaño del resto es siempre menor que la longitud del divisor, que, por lo tanto, determina el tamaño del resultado. La definición de un CRC especifica el divisor que se utilizará, entre otras cosas. Aunque CRC se puede construir utilizando cualquier tipo de regla finita, todos los CRC de uso común emplean una base finita binaria, esta base consta de dos elementos, generalmente el 0 y 1.

La Integridad de los Datos Frente a la Codificación.

Es útil para detección de errores, pero, en condiciones de seguridad, no se puede confiar en que el CRC llegue a verificar plenamente que los datos son los correctos en caso de que se hayan producido cambios deliberados y no aleatorios.

A menudo, se piensa que sí, cuando llega un mensaje, éste y su CRC coinciden, quiere decir que el mensaje no ha podido ser alterado durante su transmisión, aunque se haya transmitido por un canal abierto.

Esta suposición es falsa porque CRC es un mal método de cifrado de datos. De hecho, el CRC no se trata realmente de un método de cifrado, lo que realmente hace es utilizarse para el control de integridad de datos, pero en algunos casos se supone que se utilizarán para el cifrado.

Cuando un CRC se calcula, el mensaje se conserva (no cifrado) y la constante de tamaño CRC se sitúa hacia el final (es decir, el mensaje puede ser tan fácil como leer antes de la posición que ocupa el CRC).

Además, la longitud del CRC es por lo general mucho más pequeña que la longitud del mensaje, es imposible para una relación de 1:1 entre la CRC y el mensaje. Así múltiples códigos producirán el mismo CRC.

Por supuesto, estos códigos están diseñados para ser lo suficientemente diferentes como para variar (y por lo general sólo en uno o dos bits). Pequeños cambios en la palabra clave producirían una gran diferencia entre un CRC y otro; por ese motivo es posible detectar el error.

Si la manipulación del mensaje (cambios de los bits) es deliberada, entonces se tomará una nueva clave, produciendo un falso CRC el cual puede ser calculado para el nuevo mensaje y sustituir el CRC real en el final del paquete y esta modificación no podrá ser detectada.

Se puede concluir que la CRC sirve para verificar la integridad, pero no para saber si el mensaje es correcto.

2.4.7. Marcas de agua (Watermarking).

Una marca de agua es información oculta embebida en algunos datos. En un principio fueron desarrolladas para imágenes y audio. Se usan para proteger los derechos del propietario copyright de los datos. Los objetivos concretos de las marcas de agua son los siguientes: (14)

- *Trazabilidad de los datos:* Si se encuentra algún dato, el propietario del copyright puede comprobar con la marca de agua que ese dato le pertenece.

- *Robustez de la marca de agua:* La marca de agua no debe ser fácilmente separable; esto se refiere a la resistencia contra una lista conocida de ataques.
- *Imperceptibilidad de la marca de agua:* Si la marca de agua degrada demasiado los datos, entonces no es útil.

Existen dos clases de marcas de agua en el código:

Marcas de agua estáticas: Hay dos posibilidades, en las cuales se pueden usar los algoritmos tradicionales de marcas de agua:

- *Data watermarks:* están embebidas dentro del texto, las imágenes contenidas en el código.
- *Code watermarks:* La marca de agua se encuentra embebida en el orden de las instrucciones del código.

Marcas de agua dinámicas: Explotan una propiedad específica del código: que es ejecutable.

- *Huevos de pascua (Easter eggs):* se modifica el comportamiento del código. (Ejemplo: para determinadas combinaciones de teclas, se realizan acciones ocultas).
- *Estructuras de datos:* La marca de agua está embebida en el estado de la aplicación (variables stack, etc.) como si se estuviera ejecutando con una entrada particular.
- *Traza de ejecución:* La marca de agua está embebida dentro del orden de las instrucciones ejecutadas.

Actualmente en la práctica los algoritmos de marcas de agua deben analizarse dentro del entorno de trabajo del sistema y de acuerdo con la aplicación donde será utilizado, es por ello que se pueden analizar algunas de las posibles aplicaciones de las marcas de agua y sus peculiaridades. Ejemplo de ello es su aplicación en el:

- **Control de copias.**

  Las marcas de agua diseñadas para el control de copias, contendrán la información determinada por su propietario, acerca de las reglas de uso y copiado de los archivos en los que se insertan. A diferencia de otras marcas de agua las marcas de agua usadas en el control de copias restringen la utilización de los archivos de acuerdo con las reglas de uso y copiado que porten.

A pesar del esfuerzo de la comunidad científica y de la industria en desarrollar y establecer una tecnología de marcas de agua, hay que decir que desde el punto de vista científico y tecnológico existen numerosas incógnitas que están por resolver, muchos de los fundamentos teóricos utilizados no son totalmente concluyentes y la mayoría de los sistemas se diseñan de forma heurística.

Otro inconveniente, es la carencia de un conjunto completo de normas para evaluar los sistemas de marcas de agua, lo que puede conducir a establecer como estándar, un sistema que falle de manera espectacular, desacreditando a toda la tecnología basada en ellos.

Hoy por hoy las marcas de agua se presentan como una herramienta que puede ayudar a combatir y en todo caso a entorpecer, la proliferación de los delitos informáticos. Las aplicaciones relacionadas con la protección de los derechos de copyright, así como las pruebas de propiedad, tienen que evolucionar mucho para brindar los servicios que se esperan de ellas.

2.4.8. Packers/Protectors.

Se define packer/protector como un programa que toma como entrada un fichero ejecutable y devuelve otro fichero ejecutable con la misma funcionalidad, pero con ciertas protecciones añadidas que dificultan su análisis. Existe también la posibilidad de proteger el programa partiendo desde el código fuente e indicando las zonas calientes. (15)

Se puede decir que los primeros programas de este estilo únicamente pretendían reducir el tamaño del ejecutable, de ahí el nombre de packers, pero se fueron introduciendo técnicas de protección, dando el paso a los protectors.

¿Cómo funciona?

El packer debe cifrar al menos la sección ejecutable del fichero a proteger, y en el momento de ejecutarlo debe ser capaz de descifrarlo y ponerlo en memoria. Existen packers que cifran el fichero completo, pero la mayoría optan por introducir una nueva sección, escribir en ella su rutina de descifrado y redirigir el punto de entrada a la misma.

Por supuesto, existen diferentes maneras de implementarlo, ya que se pueden encontrar con cifrados manuales que cifran una sección con un XOR, los que introducen la rutina de descifrado en alguna parte del ejecutable no utilizada y modifican la cabecera PE para que apunte a dicha rutina.

Si se analizan las secciones de un mismo programa con y sin upx, se puede ver cómo mientras uno tiene el punto de entrada en la sección de código el otro tiene las secciones renombradas y el punto de entrada no apunta a la primera sección. El código que se ejecute en este segundo caso es el encargado de poner en memoria el programa desempacado antes de ejecutarlo.

Una protección de este tipo tiene la ventaja de que el desarrollador se despreocupa de la protección del software, y alguien ducho en el tema es el encargado de protegerla.

Cabe destacar que hoy día también se utiliza la virtualización como protección, ya que añade un grado de complejidad al análisis del programa.

Para intentar evitar que se pueda llegar a detener la ejecución en el punto de entrada original del programa (OEP), se emplean "trucos" anti-debug[3]. Además para evitar que del volcado se obtenga un fichero funcional, se emplean técnicas anti-dump[4].

2.4.9. Compresión de los datos.

La compresión de datos consiste en la reducción del volumen de información tratable (procesar, transmitir o grabar). En principio, con la compresión se pretende transportar la misma información, pero empleando la menor cantidad de espacio.

El espacio que ocupa una información codificada (datos, señal digital, etc.) sin compresión es el cociente entre la frecuencia de muestreo y la resolución. Por tanto, cuantos más bits se empleen mayor será el tamaño del archivo. No obstante, la resolución viene impuesta por el sistema digital con que se trabaja y no

[3] Técnicas que utiliza el software para detectar la presencia extraña y evitar así poder ser traceado. Normalmente intentará detectar la presencia de un debugger y si es así parar la ejecución del software mientras este esté presente.

[4] Acción legal destinada a proteger los mercados internos de la competencia desleal proveniente del exterior.

se puede alterar el número de bits a voluntad; por ello, se utiliza la compresión, para transmitir la misma cantidad de información que ocuparía una gran resolución en un número inferior de bits.

La compresión de datos se basa fundamentalmente en buscar repeticiones en series de datos para después almacenar solo el dato junto al número de veces que se repite. Así, por ejemplo, si en un fichero aparece una secuencia como "AAAAAA", ocupando 6 bytes se podría almacenar simplemente "6A" que ocupa solo 2 bytes, en algoritmo RLE.

En realidad, el proceso es mucho más complejo, ya que raramente se consigue encontrar patrones de repetición tan exactos (salvo en algunas imágenes). Se utilizan algoritmos de compresión:

- ✓ Por un lado, algunos buscan series largas que luego codifican en formas más breves.
- ✓ Por otro lado, algunos algoritmos examinan los caracteres más repetidos para luego codificar de forma más corta los que más se repiten.
- ✓ Otros construyen un diccionario con los patrones encontrados, a los cuales se hace referencia de manera posterior.

A la hora de hablar de compresión hay que tener presentes dos conceptos:

1. Redundancia: Datos que son repetitivos o previsibles.

2. Entropía: La información nueva o esencial que se define como la diferencia entre la cantidad total de datos de un mensaje y su redundancia.

La información que transmiten los datos puede ser de tres tipos:

1. Redundante: información repetitiva o predecible.

2. Irrelevante: información que no se puede apreciar y cuya eliminación por tanto no afecta al contenido del mensaje. Por ejemplo, si las frecuencias que es capaz de captar el oído humano están entre 16/20 Hz y 16.000/20.000 Hz s, serían irrelevantes aquellas frecuencias que estuvieran por debajo o por encima de estos valores.

3. Básica: la relevante. La que no es ni redundante ni irrelevante. La que debe ser transmitida para que se pueda reconstruir la señal.

Teniendo en cuenta estos tres tipos de información, se establecen tres tipologías de compresión de la información:

1. Sin pérdidas reales: es decir, transmitiendo toda la entropía del mensaje (toda la información básica e irrelevante, pero eliminando la redundante).

2. Subjetivamente sin pérdidas**:** es decir, además de eliminar la información redundante se elimina también la irrelevante.

3. Subjetivamente con pérdidas: se elimina cierta cantidad de información básica, por lo que el mensaje se reconstruirá con errores perceptibles, pero tolerables (por ejemplo: la videoconferencia).

Diferencias entre compresión con y sin pérdida:

- Compresión sin pérdida: los datos antes y después de comprimirlos son exactos en la compresión sin pérdida. En el caso de la compresión sin pérdida una mayor compresión solo implica más tiempo de proceso. El bit-rate siempre es variable en la compresión sin pérdida. Se utiliza principalmente en la compresión de texto.
- Un algoritmo de compresión con pérdida puede eliminar datos para reducir aún más el tamaño, con lo que se suele reducir la calidad. En la compresión con pérdida el bit-rate puede ser constante o variable. Hay que tener en cuenta que una vez realizada la compresión, no se puede obtener la señal original, aunque sí una aproximación cuya semejanza con la original dependerá del tipo de compresión. Se utiliza principalmente en la compresión de imágenes, videos y sonidos.

2.5. Métodos de protección basados en Hardware.

2.5.1. Hard-key, hard-lock, dongle o mochila.

Es un dispositivo electrónico que se conecta al puerto paralelo o serial de una PC. Algunos se pueden conectar a un puerto USB, y disponen de un procesador interno, lo que permite mejoras de seguridad frente a los anteriores. Para cada formato existen llaves con diferentes funcionalidades, y por lo tanto diferentes costos, que permiten trabajar en forma mono usuario o red, controlar una o varias aplicaciones, con más o menos memoria y versiones que además controlan una fecha tope.

Un hard-key es una llave de hardware que contiene código y un password que se utilizan para poder controlar el acceso a aplicaciones de software. La protección se logra incluyendo dentro de él un programa protegido con una serie de tests de validación, consultas o bloqueos. El software protegido accede a él cada cierto tiempo para verificar su presencia, en cuyo caso recibe la respuesta correcta a dicha consulta, o para leer/escribir datos en su memoria.

Básicamente contienen, dependiendo del modelo, los siguientes componentes:

- Número de serie, empleado como control interno.
- Contraseña, asignada por el fabricante al realizar el primer pedido.
- Memoria programable (de 0 a 496 bytes).
- Fecha y hora, también programables.
- Funciones de codificación/decodificación.

Como su nombre lo indica, un hard-key es una llave de acceso a aplicaciones y datos. El hard-key utiliza un algoritmo único, que es diferente para cada modelo; básicamente transforma la cadena de caracteres en una respuesta numérica, cuyo resultado se devuelve al programa que lo solicitó para evaluación y validación.

Si el hard-key correcto no es detectado, el programa dejará de funcionar. Como siempre, la efectividad de este sistema de protección depende de la sofisticación del mecanismo de bloqueo creado dentro del software. Este puede ser débil o fuerte dependiendo del nivel de protección requerido.

Además de la protección de software contra acceso no autorizado, también se puede usar para autorizar el acceso a ciertas características o diferentes versiones de paquetes de software. Esto se puede lograr haciendo que el sistema responda de manera diferente dependiendo de los valores numéricos de la respuesta que recibe.

Este método cuenta, entre otras ventajas con una completa transparencia, lo que ayuda a reducir interferencia con otros hard-keys. Sin embargo, tiene desventajas tales como el hecho de que puede ser molesto de instalar y usar porque requiere dispositivos de hardware especiales. No facilita la distribución del software por medio de Internet y no es un mecanismo de protección de copias realizable para la mayoría de las aplicaciones de software.

2.5.2. Instalaciones limitadas por el medio.

En este mecanismo, la instalación del software solo se puede realizar un número limitado de veces. Requiere que el programa se instale desde un medio re-escribible. Cuando se realiza cada instalación, el programa actualiza un contador de instalación que se encuentra en el medio. Cuando se excede el límite preestablecido no se permite ninguna instalación adicional.

Para asegurar este mecanismo de protección, se debe encriptar el archivo que contiene el contador de instalaciones, para que no sea fácil de localizar y modificar. Además, el medio re-escribible, tal como un disco, debe ser difícil de copiar, se debe fabricar de manera que contenga una firma que lo identifique unívocamente.

Este método cuenta con algunas ventajas como son: primero que no se necesita un password y segundo, previene que el usuario localice o modifique el contador de instalaciones. De igual forma se pueden citar como desventajas la existencia de problemas con la compilación de una instalación por licencias debido al mecanismo que se usa para que el disco sea difícil de copiar. Requiere un tipo de disco específico tal como una unidad de diskette y no soporta distribución del software basada en Internet.

2.5.3. Smart-Cards.

A diferencia de los Hard-key, que son específicos de un programa y una computadora, los dispositivos Smart-Card son solo específicos del usuario y genéricos para cualquier PC.

Pueden ser dispositivos activos o pasivos. Si es pasivo significa que el smart-card es sólo un almacenamiento seguro, y si es activo significa que contiene un procesador. En la industria se llaman tarjetas de memoria y tarjetas de chip respectivamente.

Dentro de una tarjeta Chip se puede ejecutar alguna parte de los programas, o desencriptar el programa en ejecución que se almacena encriptado en la computadora.

Existen las siguientes técnicas:

- Almacenar claves secretas en el smart-card.
- Almacenar reglas de uso dentro de las smart-card.
- Almacenar y ejecutar partes claves del programa.

- Desencriptar en tiempo de ejecución el programa encriptado.

Sin embargo, pueden surgir algunos problemas cuando se ejecutan múltiples programas porque se requieren múltiples smart-card. Esto se puede solucionar haciendo que un smart-card pueda delegar una licencia a otro, y luego que borre su clave privada y se bloquee a sí mismo. De esta manera, el usuario puede tener solo un smart-card y no es posible que use el anterior para ejecutar dos veces el mismo software.

2.5.4. Protección por Floppy-Checks o CD-Checks.

Estos sistemas de protección comprueban que el CD original (o diskette) del software se encuentre en la unidad de CD-ROM cada vez que el software se ejecuta. Es importante notar que estos sistemas no evitan que el CD-ROM pueda ser duplicado, lo que significa que si se introduce una copia, el software funcionará correctamente. (16)

Como método de detección cuenta con tres tipos:

• ***Control de existencia de un archivo*****:** el software controla cuando se inicia el programa o cada cierta cantidad de segundos, la presencia de archivos en el disco. Si estos no se encuentran, el programa finaliza su ejecución.

• ***Control por la etiqueta del CD*****:** si la etiqueta no coincide con la especificada, el programa finaliza.

• ***Control de número de serie del disco*****:** la aplicación verifica que el número de serie del CD coincida con el número de serie registrado. Si bien soluciona el problema de las copias del CD original (ya que en teoría los números de serie son únicos), implica la recompilación del código de control por cada copia distribuida en CD; es decir, el número de serie del CD al que se va a copiar debe ser el mismo que está registrado dentro del código del programa.

Entre sus principales desventajas se pueden citar la generación de posibles daños en el CD original por manipulación constante y la existencia de dos técnicas principales para vulnerar este sistema:

- Anticiparse a la detección de la unidad de CD.

- Anticipándose a la detección de la etiqueta de la unidad de CD.

2.6. Comparación entre algunos de los métodos de protección basada en software y protección basada en hardware.

Después de realizado el análisis de cada uno de los métodos de protección y de estudiar sus ventajas y desventajas se definieron tres de ellos para desarrollar la estrategia los cuales se basan en protección mediante software. Se descartaron los que desarrollan la protección mediante hardware ya que su característica principal es que emplean dispositivos externos los cuales tienen un costo elevado, esto impide su utilización por el Departamento de Software Médico Imagenológico.

Es por ello que los escogidos fueron: protección mediante número de serie o palabra clave, protección por tiempo y encriptación de datos; ya que son los más factibles para lograr la realización de las dos fases con que contará la estrategia que se va a proponer. Con el fin de garantizar un mejor entendimiento de por qué se escogieron estos métodos se realizó una comparación entre algunos de los que se basan en protección mediante software y los que la realizan mediante hardware. Para más información ver la tabla que se muestra a continuación.

Nombre del método	Basado en:	Principal fortaleza	Principal debilidad
Protección por número de serie o palabra clave.	**Protección mediante Software.**	Se pueden diseñar, implementar y codificar programas e incorporar la protección más tarde.	Es simple de vencer mediante utilidades como debuggers.
Protección por tiempo.	**Protección mediante Software.**	Ofrece la posibilidad de hacer versiones de libre distribución, para que el cliente pruebe el programa.	Se debe dejar una marca en la PC para saber si ya expiró el programa que se está protegiendo.
Encriptación de datos.	**Protección mediante Software.**	**Simétricos:** -Simplicidad y	**Simétricos:** -Dificultad de

		velocidad. **Asimétricos:** -Encriptan con una llave pública y desencriptan con una privada lo que genera mayor seguridad.	almacenar y proteger claves diferentes. **Asimétricos:** -Son muy lentos y no pueden encriptar cantidades de información.
Ofuscación de datos.	**Protección mediante software.**	Dificulta la realización de ingeniería inversa sobre el código.	Puede ser vulnerada mediante el uso de análisis estáticos y dinámicos y puede descuidar otros aspectos del software mientras se ejecuta.
Hard-key, hard-lock, dongle o mochila.	**Protección mediante Hardware.**	Protege código fuente valioso, información de texto sensible, y archivos de datos.	Tiene un costo de uso adicional muy elevado.
Protección por Floppy-Checks o CD-Checks.	**Protección mediante Hardware.**	Se utilizan para evitar versiones recortadas del CD original.	Utilización ineficiente de los recursos de la computadora.
Instalaciones limitadas por el medio.	**Protección mediante Hardware.**	Encripta el contador de ejecuciones lo que previene que el usuario localice o modifique el contador.	Requiere un tipo de disco específico y no soporta distribución del software basada en Internet.

Tabla 1. Comparación entre los métodos de protección basada en software y protección basada en hardware.

Conclusiones.

En este capítulo se describieron cada uno de los métodos de protección de software. También se hizo referencia a temas relacionados como ventajas y desventajas de los mismos, llegándose a la conclusión de que todas las técnicas de protección existentes en la actualidad tienen vulnerabilidades. Sin embargo, se puede trazar una estrategia de protección agrupando algunas de ellas para lograr una mayor seguridad de los productos que se desarrollen.

CAPÍTULO 3: CARACTERÍSTICAS DE LOS SOFTWARE PRODUCIDOS POR EL DEPARTAMENTO DE SOFTWARE MÉDICO IMAGENOLÓGICO DEL CESIM.

En este capítulo se realizará una pequeña descripción de los productos que han sido desarrollados en el Departamento de Software Médico Imagenológico, así como un análisis de las principales tecnologías y herramientas que fueron utilizadas para su creación. Además, se estudiarán las necesidades de protección de los mismos para así conformar una estrategia que sea la más adecuada.

3.1. Descripción de los software desarrollados en el Departamento de Software Médico Imagenológico.

La creación del producto alas PACS-alas RIS tiene como objetivo principal ofrecer al personal médico que labora en los Departamentos de Diagnóstico por Imágenes una amplia gama de herramientas de propósito general que permitan la visualización, el procesamiento de imágenes médicas y la edición de los informes que son emitidos.

Puede facilitar además el acceso a las imágenes desde cualquier punto de la institución de salud, el intercambio de imágenes entre unidades médicas y la creación de las listas de trabajo para los equipos de adquisición de imágenes DICOM compatibles. Dicha solución está constituida por un conjunto de sistemas altamente integrados y compatibles con el estándar internacional DICOM 3.0, que pueden ser instalados por separados:

- **alas PACSViewer**: Estación de diagnóstico general.
- **alas PACSServer**: Servidor de imágenes médicas.
- **alas PACSWorklist**: Servidor de listas de trabajo.
- **alas PACSReport**: Herramienta de edición de informes imagenológicos.
- **alas PACSDICOMAIL**: Sistema para el intercambio de imágenes médicas entre instituciones hospitalarias.
- **alas RIS:** Sistema de Información Radiológica para el manejo de la información asociada a los pacientes y los informes generados.

3.2. Descripción general de los sistemas que integran el producto alas PACS-alas RIS.

3.2.1. Estación de diagnóstico general, alas PACSViewer.

Posee herramientas para el procesamiento, análisis y visualización de las imágenes médicas con funcionalidades básicas y de post procesamiento 3D. Permite la conexión remota desde las estaciones de trabajo hasta el servidor de imágenes de la institución, recibe los estudios directamente de los equipos de generación de imágenes, intercambia estudios entre las estaciones de diagnóstico y genera informes imagenológicos. Además, posibilita que se puedan exportar a formatos comunes de imágenes, videos digitales y la impresión de imágenes en papel o películas radiográficas. Es una aplicación de escritorio.

3.2.2. Herramienta para la edición de informes: alas PACSReport.

El Reportador es una aplicación de escritorio. Constituye un importante componente del sistema alas PACSViewer. Su objetivo principal es construir un informe del estudio médico imagenológico realizado a un paciente, o modificar uno que se encuentre de forma remota, en el servidor de informes del sistema alas RIS.

Este módulo del sistema funciona en los diferentes servicios de imágenes de la clínica. Su modo de funcionamiento puede ser: integrado al visor, para los especialistas que generan sus propios informes o como una herramienta independiente para las secretarias de transcripción que reescriben los informes generados por los especialistas.

3.2.3. Módulo servidor de imágenes alas PACSServer.

Posibilita la gestión de la información de los estudios que se generan en las diferentes modalidades diagnósticas, soporta asociaciones simultáneas, así como garantizar el archivo de cada uno de estos estudios de forma ordenada.

Este módulo posibilita además la búsqueda y recuperación de los estudios desde cualquier estación de trabajo o equipo de generación de imágenes. Adicionalmente el servidor cuenta con un grupo de herramientas para la administración de los recursos del servidor, así como la sincronización de la información que hay en las bases de datos y el archivo físico y la ejecución de tareas programadas ante situaciones críticas. Es una aplicación de escritorio.

3.2.4. Servidor de listas de trabajo alas PACSWorklist.

Permite la recepción de la información de las citas de los pacientes del departamento de radiología creados en un sistema RIS ó HIS, almacenarlas y con estas conformar las listas de trabajo de los equipos de adquisición; comunicarse con los equipos de adquisición de imágenes DICOM compatibles y enviarles las listas de trabajo para una petición determinada. Mantiene informado al sistema RIS ó HIS del trabajo realizado por los equipos y el estado de la realización de los estudios. Es una aplicación de escritorio.

3.2.5. Módulo de intercambio de imágenes médicas alas PACSDICOMail.

Permite el envío de las imágenes médicas entre especialistas de diferentes unidades médicas, así como dentro de la misma institución, siendo esta una herramienta para facilitar las consultas de segunda opinión entre especialistas.

La solución utiliza la experiencia de usuarios en soluciones de mensajería lo cual facilita el uso de la misma por parte de los especialistas. Esta solución cuenta con dos herramientas, una visible a los especialistas para el envío y recepción de las imágenes y una solución servidora que gestiona los diferentes tipos de envíos hacia dentro o fuera de las instituciones.

No supone por parte de los especialistas de un conocimiento previo de técnicas sofisticadas de la rama de la informática ya que la solución que brinda es en extremo sencilla y forma parte del hacer cotidiano de cualquier persona ligada a la informática. Le da además un valor agregado al PACS, pues hace posible y extremadamente sencilla la consulta de segunda opinión, abriendo los caminos así a la Telemedicina. Es una aplicación de escritorio.

3.2.6. Sistema de Información Radiológica alas RIS.

Permite el registro de pacientes y sus citas para estudios o consultas de imagenología, el registro de los datos de los especialistas y los equipos médicos. Permite la personalización mediante perfiles de usuario, es altamente configurable y es adaptable a las condiciones particulares de las instituciones hospitalarias.

Facilita el control de una historia clínica imagenológica, así como las salidas de la estadísticas médicas y las hojas de cargo. Posee un servidor de listas de trabajo DICOM compatible que se comunica con los equipos para que estos actualicen sus listas de trabajo o para especialistas, y permite realizar búsquedas por pacientes, estudios y diagnósticos médicos, facilitando la realización de estudios de morbilidad. Es una aplicación web.

3.3. Descripción de la herramientas y tecnologías utilizadas para la realización del producto alas PACS/alas RIS.

Para Windows:

Herramientas:

- Se utilizó el Windows XP ServicePACK3 como sistema operativo el cual constituye una versión de Microsoft Windows, línea de sistemas operativos desarrollado por Microsoft. SP3 contiene nuevas características: actualizaciones independientes de Windows XP y características tomadas de Windows Vista. El SP3 puede ser instalado en las versiones *retail* y OEM de Windows XP y tener funcionalidad completa durante 30 días sin necesidad de introducir una clave de producto. Pasado ese tiempo, se le pedirá al usuario que introduzca una clave válida y active la instalación. Las versiones de tipo licencia por volumen (VLK) necesitan también que se introduzca una clave de producto.
- Para el control de versiones de ficheros se empleó el software Tortoise 1.6.un SCM fácil de utilizar y, posiblemente, el mejor cliente de Subversion independiente que hay. Está implementado como una extensión del shell de Windows, lo que hace que se integre perfectamente en el Explorador de Windows. Como no es una integración de un IDE específico el usuario puede utilizarlo como herramienta de desarrollo.

- El control de versiones fue facilitado por el VisualSVN-1.7.7.que es un cliente de Subversion, implementado como un addin para Microsoft Visual Studio, que proporciona una interfaz para realizar las operaciones más comunes de revisión de control directamente desde dentro del Visual Studio IDE. Utiliza TortoiseSVN para ejecutar los comandos de Subversion. (17)

- EMS PgManager 2009 es una herramienta de alto rendimiento para la administración de bases de datos PostgreSQL que también fue empleada en el desarrollo del producto alas PACS/alas RIS.

- Moma que constituye una herramienta provista por la plataforma .NET del proyecto MONO que brinda la posibilidad de analizar si se pueden realizar las migraciones de un producto a diversas plataformas.

- Enterprise Architect 7.5.se usó para el modelado en UML de la solución bajo la plataforma Windows.
- Visual Studio 2008 Team System fue la herramienta de desarrollo que sirvió para la realización del software. Permite la creación de soluciones multiplataforma adaptadas para funcionar con las diferentes versiones de .Net Framework: 2.0. (Incluido con Visual Studio 2005), 3.0 (incluido en Windows Vista) y 3.5 (incluido con Visual Studio 2008).
- Framework .NET 3.5 incluye biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web.
- Postgres 8.4. es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Tecnología:

- Windows Presentation Fundation que es un modelo de programación unificado para generar experiencias de clientes inteligentes de Windows, en las que se incorpora la interfaz de usuario, multimedia y documentos.

Para Linux

Herramientas

- OpenSUSE 11.2. Como última versión de un proyecto libre auspiciado por Novell y AMD para el desarrollo y mantenimiento de un sistema operativo basado en Linux.
- Mono 2.6. Es una plataforma de desarrollo para aplicaciones en múltiples lenguajes de programación, incluyendo entre ellos Python, Object Pascal, Nermele, y C#. Es un proyecto de código abierto impulsado por Novell, como una implementación del .NET Framework de Microsoft y del estándar European Computer Manufacturers Association (ECMA). Licenciado bajo la GPL y actualmente funciona en GNU/Linux, Free BSD, UNIX, Mac OS X, Solaris y plataformas Windows.
- Monodevelop 2.2. Como entorno de desarrollo (IDE) para Mono y Gtk # con todas las funciones integradas. Originariamente se trata de un puerto de SharpDevelop 0,98. (18)

- Apache 2.2 es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

- XSP2 como un servidor web sencillo que ejecuta de forma independiente y está escrito en C#. El mismo permite hospedar en Linux y otros sistemas operativos UNIX sitios desarrollados utilizando ASP.NET. Además de ejecutarse sobre la plataforma Mono para Linux también permite su ejecución sobre la plataforma .NET, posibilitando que sea utilizado como un servidor web ligero en cualquier plataforma que soporte .NET.

- Postgres 8.4.como un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Tecnología:

- GTK #.Biblioteca del equipo GTK+, la cual contiene los objetos y funciones para crear la interfaz gráfica de usuario. Maneja widgets como ventanas, botones, menús, etiquetas, deslizadores, pestañas, etc.

3.4. Necesidades de protección.

Entre las necesidades de protección que hoy se presentan en el Departamento de Software Médico Imagenológico se pueden mencionar:

1. La inexistencia de los demos de prueba como un mecanismo para afianzar a los clientes.
2. Garantizar la protección máxima de las aplicaciones contra el uso ilegal.
3. La protección del código fuente.
4. Garantizar la integridad de la instalación de los software.
5. Un mecanismo para identificar los software de acuerdo con sus licencias.

Para ello surge la necesidad de desarrollar una estrategia que se pueda reutilizar como esquema de licenciamiento no solo para protección sino también para el soporte que se da a las aplicaciones, dígase actualizaciones automáticas, servicios de solución de problemas que puedan presentarse, entre otros.

Conclusiones.

En este capítulo se analizó y comprobó que cada sistema ofrece una solución capaz de adaptarse a los requerimientos de los distintos tipos de unidades médicas. Poseen una completa integración entre los diferentes módulos, además de una interfaz gráfica común que facilita su comprensión y uso y permiten la gestión de forma eficiente de las actividades relacionadas con el trabajo en los Departamentos de Diagnóstico por Imágenes Médicas, y la información asociada.

Ofrecen además la opción de poder entregar a los pacientes CD/DVD con el estudio practicado para que sea revisado con posterioridad por los clínicos. Todo lo expuesto anteriormente debe estar asegurado por lo que se vuelve necesario desarrollar una estrategia de protección como la que se propondrá en el siguiente capítulo.

CAPÍTULO 4. ESTRATEGIA PARA EL LICENCIAMIENTO DE LOS SOFTWARE DESARROLLADOS POR EL DEPARTAMENTO DE SOFTWARE MÉDICO IMAGENOLÓGICO DEL CESIM.

Posterior al estudio de los capítulos anteriores, donde se presentan la fundamentación teórica de la investigación, los métodos de protección existentes y las características de los software que son desarrollados en el departamento se puede dar paso a la descripción del cuarto capítulo donde se abordarán temas como: la metodología de la estrategia de protección que se propondrá, la descripción de la estrategia y la validación y aprobación de las herramientas con las que se desarrollará la misma.

4.1. Línea de vida del software analizada para el desarrollo de la estrategia propuesta.

Se definió la línea de vida del software la cual está dividida en dos partes fundamentales. La primera es la elaboración donde se lleva a cabo la implementación del esquema de licenciamiento por parte de los desarrolladores, así como la definición de las herramientas a utilizar por los mismos. La otra parte es la etapa de explotación en la que se prueba cómo debe funcionar el esquema desarrollado desde el punto de vista del técnico de despliegue y de los usuarios, es decir, aquí se ajustan las sub-etapas: instalación y demo. La versión full operativa queda fuera del alcance de la investigación.

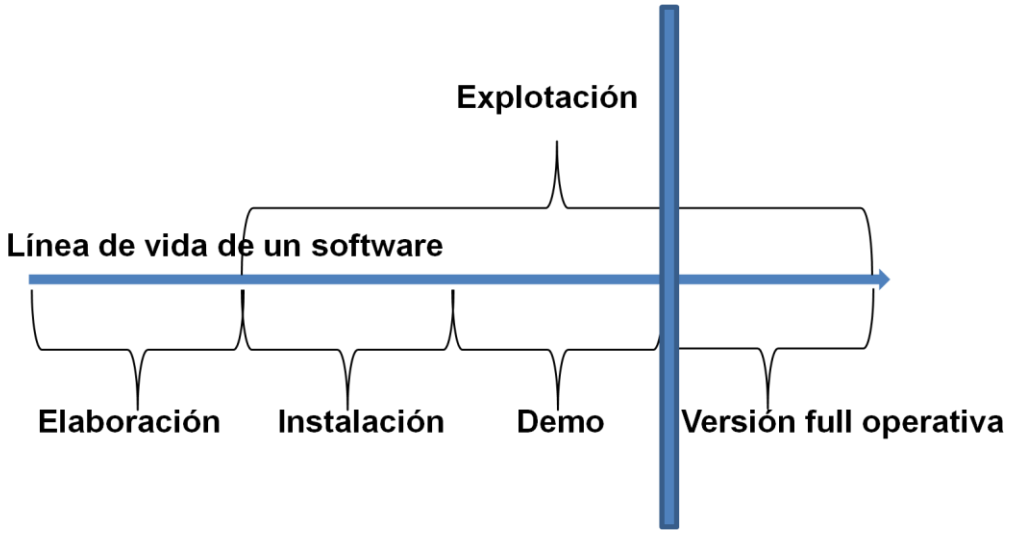

Figura2: Línea de vida del software.

4.2. Estrategia de protección de software.

La estrategia de protección de software para el Departamento de Software Médico Imagenológico del CESIM cuenta con dos objetivos fundamentales para los cuales se trazarán una serie de acciones a realizar que posibiliten su cumplimiento, esto se recoge dentro del proceso que será especificado en las fases de la misma.

Se apoyará en una metodología de trabajo que estará basada en dos preguntas fundamentales: ¿Qué?, refiriéndose a las técnicas de protección y a las tecnologías que se emplearán para su desarrollo y ¿Cómo?, es decir, proceso de desarrollo y personal encargado del mismo. (19)

4.2.1. Elementos que componen la estrategia. (20)

- **Proceso:** es necesario conocer su descripción, si tiene artefactos de entrada y salida, etc.
- **Personal:** se hace vital definir los roles y proporcionar una pequeña descripción de los mismos.
- **Tecnologías:** se definirán las tecnologías y herramientas a utilizar en la estrategia.

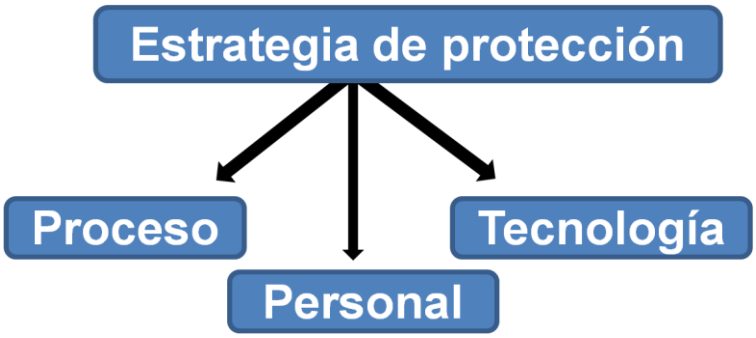

Figura 3: Elementos de la estrategia de protección.

4.3. El personal.

El personal constituye un elemento de vital importancia a la hora del desarrollo y la aplicación de la estrategia de protección. También se hace importante definir una estructura organizacional o jerárquica donde se observen las funciones de cada uno de los roles, así como sus responsabilidades en la realización de cada una de las tareas.

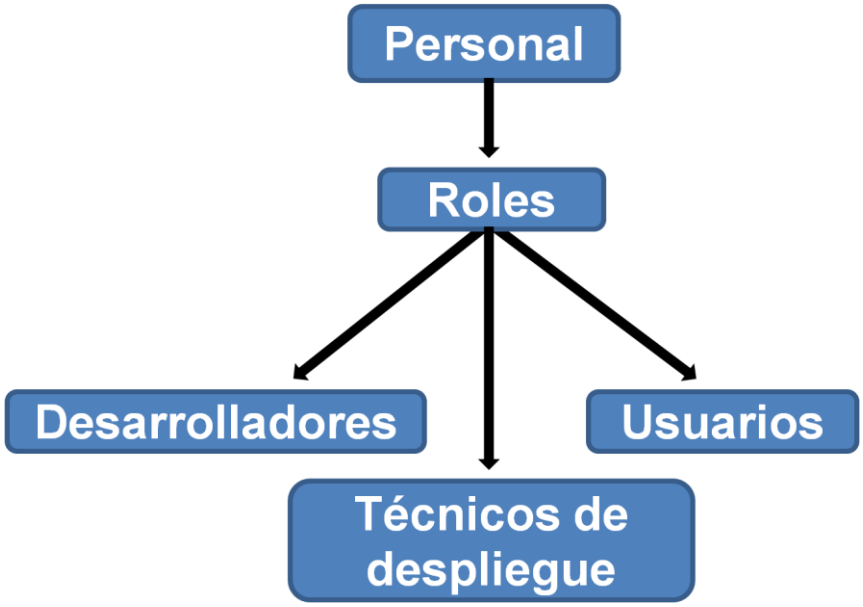

Figura 5: Personal.

4.3.1. Definición jerárquica de los roles.

Desarrolladores: Son los encargados de llevar a cabo la implementación de los métodos definidos para la realización de la estrategia. Además, realizan las pruebas para ver si la seguridad está garantizada en las dos etapas de la fase de explotación que están en el marco de esta investigación. Estas pruebas se llevan a cabo para las dos versiones de la aplicación, sea DEMO o versión full operativa. Realizan además las acciones correspondientes a la fase de soporte de la aplicación.

Técnicos de despliegue: Se encargan de llevar a cabo el proceso de instalación y mantenimiento de la aplicación.

Usuarios: Pueden ser los técnicos de despliegue o el usuario final, es decir, el comprador del producto. Realizará la instalación del software y podrá probar el producto en su versión DEMO.

4.4. Tecnologías.

La tecnología se considera un elemento imprescindible en el desarrollo de la sociedad ya que otorga grandes beneficios y ha contribuido a modificar el entorno en el que habita el hombre. En la estrategia de protección de software se necesita su aplicación a la hora de desarrollar las acciones que darán

cumplimiento a los objetivos de cada fase. Serán definidas de acuerdo con sus ventajas y desventajas. Además, debe analizarse con los directivos del departamento ya que cualquier disposición técnica debe ser previamente aprobada por ellos.

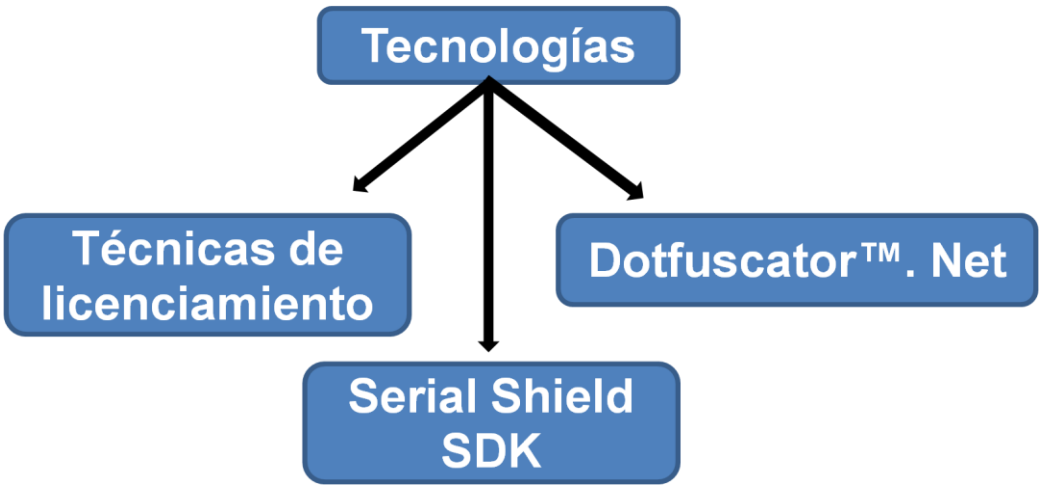

Figura 6: Tecnologías

4.4.1. Evaluación de las técnicas de licenciamiento de acuerdo con los atributos definidos que cumplen las necesidades del Departamento de Software Médico Imagenológico del CESIM.

Después de haber realizado un estudio de las técnicas de licenciamiento de software existentes y haber comparado algunas de ellas en cuanto a sus principales fortalezas y debilidades, se realizó un análisis teniendo en cuenta sus características, las que fueron descritas en el capítulo dos de la presente investigación. Esto se realizó con el propósito de determinar cuál o cuáles de ellos son los más factibles para el desarrollo de la estrategia. Para ello se tuvieron en consideración los siguientes aspectos:

Indicadores.

Regular: 0-3

Bueno: 4-6

Muy bueno: 7-10

Atributos.

1- Garantizar la protección máxima de las aplicaciones contra el uso ilegal.
2- Proteger el código fuente.
3- Garantizar la integridad de la instalación de los software.
4- Permitir la realización de versiones de prueba eficientes.
5- Flexibilidad en la distribución.

Métodos técnicos.

- **Software:**

Protección x tiempo.
Protección x número de serie o palabra clave.
Llaves de software empleando datos del hardware.
Ofuscación del código.

- **Hardware:**

Llaves de hardware.
Instalaciones limitadas por el medio.
Floppy o check disc.

Métodos de protección	Protección contra el uso ilegal.	Protección del código fuente.	Integridad en la instalación de los software.	Permita realizar versiones de prueba eficientes.	Flexibilidad en la distribución.
Protección x tiempo.	6	0	0	8	5
Protección x número de	4	0	1	5	9

serie o palabra clave.					
Llaves de software empleando datos del hardware.	10	0	10	4	4
Llaves de hardware.	10	0	6	4	6
Instalaciones limitadas por el medio.	3	0	2	0	5
Protección mediante Floppy o check disc.	2	0	2	0	5
Ofuscación.	0	9	0	0	6

Tabla 2: Evaluación de los métodos de acuerdo con los atributos determinados.

4.4.2. Resultado del análisis de la efectividad de los métodos.

Este análisis arrojó como resultado que los métodos más factibles y que mejor se adecuan al esquema híbrido que se pretende desarrollar en la estrategia son: la protección x tiempo y la protección x palabra clave o número de serie para garantizar la realización de versiones de prueba eficientes y una mayor flexibilidad en la distribución de los productos.

Para que la integridad de la instalación esté garantizada al igual que la protección contra el uso ilegal, se encontró como método más efectivo el de llaves de software empleando datos del hardware. Después de realizado este análisis se comprobó que la herramienta Serial Shield SDK implementa todos estos métodos, por lo cual se propone esta última para el desarrollo de la estrategia.

Como técnica más viable en la protección del código fuente se concluyó la ofuscación de datos. Esta última se puede implementar mediante la utilización de la herramienta Dotfuscator .NET™ el cual constituye un sistema de recompilación post desarrollo para aplicaciones .NET. El mismo analiza las aplicaciones y hace que la ofuscación que realiza para protegerlas sea más pequeña, rápida y fuerte ante la ingeniería inversa.

4.5. El proceso.

Es uno de los componentes más importantes ya que en él se describe la metodología utilizada para el desarrollo de la estrategia de protección. Agrupa los objetivos que persigue la estrategia y las acciones a realizar para dar cumplimiento a los mismos, los cuales serán especificados en la descripción de la misma. No cuenta con ningún artefacto de entrada, pero sí tiene como artefacto de salida la plantilla: Estrategia para la protección y el licenciamiento de los software del Departamento de software Médico Imagenológico del CESIM.

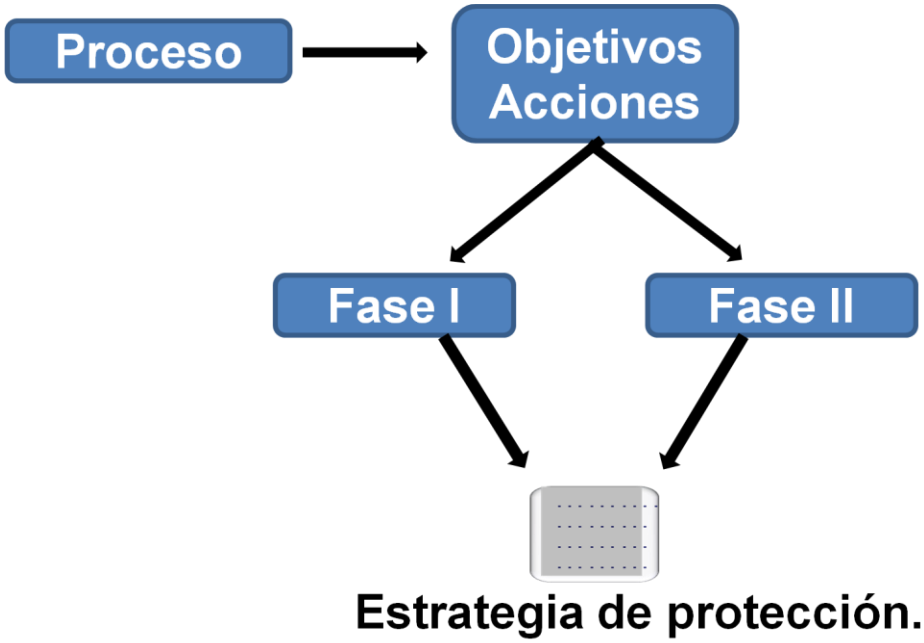

Figura 4: Proceso.

4.5.1. Descripción de la estrategia de protección.

Objetivo general de la estrategia.

Esta estrategia persigue como objetivo fundamental garantizar la protección de los software que se desarrollan en el Departamento de Software Médico Imagenológico del CESIM, así como garantizar en una medida el soporte de los mismos.

Principios de la estrategia.

- Toda la estructura de la estrategia debe estar en concordancia con lo establecido por la dirección del Departamento de Software Médico Imagenológico del CESIM.
- Las decisiones técnicas deben estar bajo la supervisión de la dirección del Departamento de Software Médico Imagenológico del CESIM

Alcance de la estrategia.

Proteger las aplicaciones que se desarrollen en el Departamento de Software Médico Imagenológico del CESIM, lo cual será implementado durante la fase de elaboración del producto que se puede apreciar en la línea de vida del software analizada para la presente investigación.

Planeación de la estrategia.

Para lograr el objetivo planteado se determinaron los objetivos de cada fase y las acciones a realizar para dar cumplimiento a cada uno de ellos. Cuando se combinan estas partes se logra la composición de la estrategia en sí, la que se mostrará a continuación.

Fase I: Elaboración.

Objetivo.

Lograr la protección de las aplicaciones contra el uso ilegal e impedir el acceso al código fuente de la aplicación empleando para ello los métodos de licenciamiento definidos.

Acciones:

Para los desarrolladores:

1. Realizarán un estudio del sistema Serial Shield SDK para luego utilizarlo en la implementación de la protección de las aplicaciones, ya que este garantiza el éxito de la mayoría de los atributos evaluados.

2. Realizarán un estudio del sistema Dotfuscator .NET™.
3. Para el desarrollo del método ofuscación de datos, como parte fundamental en la protección del código fuente de las aplicaciones se utilizará el sistema Dotfuscator .NET™.
4. Definirán el tipo de licencia de la aplicación, dígase versión trial o full versión, mediante el uso de la herramienta Serial Shield SDK.
    i. Si es versión trial, implementará la protección por tiempo de la aplicación de acuerdo con las opciones que da esa herramienta.
    ii. Para las versiones de prueba se generarán los números de serie de cada versión empleando para ello la herramienta antes mencionada.
    iii. Si es versión full desarrollarán la protección contra el uso ilegal mediante las opciones de bloqueo de la llave de activación utilizando el ID que el sistema extrae de la PC o utilizando el número de serie, lo cual garantiza también la integridad en la instalación.

Fase II: Explotación.

Objetivo.

Verificar que el esquema de licenciamiento hibrido implementado durante la fase de elaboración funcione de manera eficiente.

Acciones:

<u>Para la sub-fase instalación.</u>

Acciones:

Técnicos de despliegue:

1. Comprueban que la instalación se pueda realizar mediante el uso del número de serie que lleva impreso el CD.
    i. Si no hay número de serie no se puede instalar la aplicación.
2. Después de instalar, deben configurar la aplicación para garantizar el buen funcionamiento de la misma.
3. Activar el servicio de mensajería de la aplicación (si los usuarios tiene internet) para garantizar un mejor soporte.

Para la sub-fase DEMO.

Acciones:

Técnicos de despliegue.

1. Probarán las funcionalidades de la aplicación.
    i. Chequearán que cumpla con las condiciones de una versión de prueba, es decir, que permita el acceso a las funciones, pero de forma limitada.
2. Podrán registrar la aplicación pasando a ser una versión full.
    i. Esta permite el acceso total a todas las funciones, sin ninguna limitación.
    ii. Posibilita la total explotación de la aplicación así como el acceso a todos los servicios que ofrece la empresa desarrolladora en materia de soporte.
3. Deben brindar capacitación a los usuarios de la aplicación en los temas referidos al trabajo con la misma.

4.6. Validación de las herramientas que se utilizarán en el desarrollo de la estrategia.

Para la realización de la estrategia se utilizarán las herramientas y tecnologías que ya fueron definidas por la dirección del Departamento de Software Médico Imagenológico del CESIM, es decir, las mismas que fueron empleadas en la construcción de los sistemas que se describieron el capítulo 3 del presente Trabajo de diploma.

Se empleará además el sistema Serial Shield SDK como una herramienta para realizar el proceso del licenciamiento de las aplicaciones que se desarrollen ya que este permite generar las licencias necesarias para proteger las soluciones ya sea para versiones de prueba como para la versión full operativa. Este sistema brinda la posibilidad de agregar capacidades de evaluación a las aplicaciones y maneja todos los aspectos para asegurarlas durante el período de evaluación y que estén registradas correctamente en una máquina específica.

También se propone la utilización del sistema Dotfuscator .NET™ para desarrollar la ofuscación del código de la aplicación ya que permite enriquecer el mismo y hacerlo más seguro garantizando que tamaño de los programa sea más pequeño. Además, realiza verificaciones de integridad y mejoras en cuanto a tiempo de ejecución y habilita la aplicación mediante señales del sistema operativo.

Conclusiones.

En este capítulo se expusieron temas como: los objetivos y acciones del proceso correspondiente a la estrategia de protección que se propone, la descripción de la misma y la aplicación de la metodología de dicha estrategia. Luego de tratar estos contenidos, se puede concluir que la estrategia desarrollada para garantizar la seguridad de los software del Departamento de Software Médico Imagenológico es viable y que se puede comenzar a aplicar.

CONCLUSIONES

El presente trabajo ha atravesado diferentes etapas que han permitido dar cumplimiento al objetivo propuesto. En la etapa inicial de investigación, se realizaron búsquedas sobre temas relacionados principalmente con el proceso de licenciamiento y las técnicas de protección de software existentes, lo que permitió definir cuales se utilizarían para desarrollar el esquema de licenciamiento hibrido que se propondrá.

Posteriormente se definió la línea de vida de los software que se producen en el departamento de Software Médico Imagenológico para especificar bien las fases de desarrollo de los mismos y así lograr una mejor concepción de la estrategia.

Se propusieron las herramientas Serial Shield SDK y Dotfuscator™. Net para el desarrollo de la estrategia de protección. Estas brindan a las aplicaciones un mayor grado de seguridad e implementan las técnicas de protección que se definieron para la creación de la estrategia.

Se diseñó y elaboró la estrategia que se propone. Esta posibilita lograr una descripción detallada de cada uno de los pasos para llevar a cabo la protección de software durante su etapa de elaboración así como algunas acciones a desarrollar en las sub-fases de explotación que se encuentran dentro del alcance del trabajo.

El proceso generó como artefacto de salida la plantilla: Estrategia para la protección y el licenciamiento de los software del Departamento de Software Médico Imagenológico del CESIM, en la cual se describe específicamente la estrategia y se espera su utilización logre garantizar la máxima seguridad de las soluciones que se desarrollen.

RECOMENDACIONES

Después de terminada la estrategia de protección de software diseñada en este Trabajo de diploma se recomienda:

- Su empleo para evitar que el Departamento de software Médico Imagenológico sufra algún tipo de filtración o robo de sus productos.
- La compra del sistema Dotfuscator™. Net para su utilización en el desarrollo de la estrategia de protección.
- Profundizar más en el estudio de las técnicas de protección existentes para que puedan ser empleadas en la protección de la tercera etapa de la fase de explotación definida en la línea de vida del software que se analizó para el desarrollo de la presente investigación, ya que la versión full operativa quedó fuera del alcance de la tesis y es de vital importancia garantizar su seguridad.
- Continuar investigando sobre las posibles combinaciones de técnicas de protección para seguir fortaleciendo la seguridad de los software que se realicen, lo cual serviría como una nueva versión de la estrategia que en este trabajo se propone.
- Analizar cómo podría emplearse esta estrategia en las actividades de soporte que se brinda a las aplicaciones del departamento.

REFERENCIAS BIBLIOGRÁFICAS

1. Microsoft. [En línea]

http://www.microsoft.com/latam/softlegal/sam/what_basics_licensingWhat.mspx.

2. Culero Vazquez, Monserrate, Gomez Herrera, Wendy Guadalupe y Torres Sánchez, Susana. Bakara. [En línea] [Citado el: 2 de marzo de 2010.]

http://bakara.files.wordpress.com/2007/04/softwarelibrevssoftwarepropietario.pdf.

3. Microsoft. [En línea] http://www.microsoft.com/latam/softlegal/sam/lic_cal.mspx

4. Free Download Manager. [En línea] 18 de diciembre de 2006. [Citado el: 20 de febrero de 2010.]

5. ABOX. [En línea] [Citado el: 7 de junio de 2010.]

http://www.abox.com/productos.asp?pid=494.

6. Microsoft. [En línea] 7 de abril de 2008. [Citado el: 20 de enero de 2010.]

http://www.microsoft.com/spain/sharedsource/licensingbasics/licensingmodels.mspx.

7. Transtecnia. [En línea] http://www.transtecnia.cl/contrato_mantencion.htm.

8. Mercadeo.com. [En línea] http://www.mercadeo.com/63_saas.htm.

9. Cors, Israel y Pernich, Patricia. criptored.es. [En línea] [Citado el: 21 de febrero de 2010.]

http://www.criptored.upm.es/guiateoria/gt_m148p.htm .

10. Rubio Rincón, Jaime Hernando. escuelaing.edu.cu. [En línea] 2005.

http://profesores.is.escuelaing.edu.co/asignaturas/sypi20071/FOLLETOS%20Y%20MATERIAL%20DE%20ESTUDIO/La%20protecci%F3n%20del%20software.pdf.

11. Siliconkey. [En línea] [Citado el: 2 de marzo de 2010.]
http://www.siliconkey.com.ar/time_protect.htm.

12. Olivares, Juan y Barajas, Sandokan. sg.com. [En línea] [Citado el: 3 de marzo de 2010.]

http://www.sg.com.mx/content/view/767.

13. Wapedia. [En línea] [Citado el: 4 de marzo de 2010.]

http://wapedia.mobi/es/Control_de_redundancia_c%C3%ADclica.

14. Orue Lopez, Amalia Beatris. iec.csic.es. [En línea] [Citado el: 4 de marzo de 2010.]

http://www.iec.csic.es/CRIPTonOMICon/articulos/expertos88.html.

15. blogs.s21sec. [En línea] [Citado el: 20 de febrero de 2010.]
http://blog.s21sec.com/2008/10/proteccin-del-software-parte-v.html.

16. Silver, Mr. Fortunecity. [En línea] [Citado el: 3 de marzo de 2010.]

17. TortoiseSVN.com. [En línea] http://tortoisesvn.net/.

18. MonoDevelop.com. [En línea] http://monodevelop.com/Download.

19. Wordpress.com. [En línea]
http://ejecucion.wordpress.com/2008/10/09/pasos-para-hacer-estrategia-parte-i/.

20. Trabajo.com. [En línea] http://www.trabajo.com.mx/vision_de_una_empresa.htm.

BIBLIOGRAFÍA

abcdatos.com. [En línea] [Citado el: 20 de febrero de 2010.]

http://www.abcdatos.com/tutoriales/tutorial/z4381.html.

ABOX. [En línea] [Citado el: 7 de junio de 2010.]

http://www.abox.com/productos.asp?pid=494.

Agilizate.com. [En línea] http://www.agilizate.com/software-en-alquiler-que-es.html.

bsa.org. [En línea] http://w3.bsa.org/latinamerica/antipiracy/Why-a-License-Matters.cfm.

Argos. [En línea]

http://www.argos.com.ec/index.php?option=com_content&view=article&id=49&Itemid=62.

blogs.s21sec. [En línea] [Citado el: 22 de febrero de 2010.]

http://blog.s21sec.com/2008/09/proteccin-del-software-parte-ii.html.

blogs.s21sec. [En línea] [Citado el: 20 de febrero de 2010.]

http://blog.s21sec.com/2008/10/proteccin-del-software-parte-v.html.

Business Software Alliance.[En línea]

http://w3.bsa.org/latinamerica/antipiracy/Why-a-License-Matters.cfm.

Ciberaula.com. [En línea] http://linux.ciberaula.com/articulo/linux_apache_intro/.

content4reprint.com. [En línea] http://www.content4reprint.com/view/spanish-108252.htm .

Cors, Israel y Pernich, Patricia. criptored.es. [En línea] [Citado el: 21 de febrero de 2010.]

http://www.criptored.upm.es/guiateoria/gt_m148p.htm .

Culero Vazquez, Monserrate, Gomez Herrera, Wendy Guadalupe y Torres Sánchez, Susana. Bakara. [En línea] [Citado el: 2 de marzo de 2010.]

http://bakara.files.wordpress.com/2007/04/softwarelibrevssoftwarepropietario.pdf.

Free Download Manager. [En línea] 18 de diciembre de 2006. [Citado el: 20 de febrero de 2010.] http://www.freedownloadmanager.org/es/downloads/.

Google.com. [En línea] http://www.google.com/support/youtube/bin/answer.py?hl=es&answer=83734.

Google.com. [En línea] http://www.google.com/support/youtube/bin/answer.py?hl=es&answer=83734.

Mercadeo.com. [En línea] http://www.mercadeo.com/63_saas.htm.

Microsoft. [En línea] http://www.microsoft.com/latam/softlegal/sam/what_basics_licensingWhat.mspx.

Microsoft. [En línea] http://www.microsoft.com/latam/softlegal/sam/lic_cal.mspx.

Microsoft. [En línea] 7 de abril de 2008. [Citado el: 20 de enero de 2010.] http://www.microsoft.com/spain/sharedsource/licensingbasics/licensingmodels.mspx.

Microsoft.com. [En línea] http://www.microsoft.com/spain/licencias/introduccion.mspx.

MonoDevelop.com. [En línea] http://monodevelop.com/Download.

Olivares, Juan y Barajas, Sandokan. sg.com. [En línea] [Citado el: 3 de marzo de 2010.] http://www.sg.com.mx/content/view/767.

Orue Lopez, Amalia Beatris. iec.csic.es. [En línea] [Citado el: 4 de marzo de 2010.] http://www.iec.csic.es/CRIPTonOMICon/articulos/expertos88.html.

Pereira, Jorge E. Mercadeo.com. [En línea] http://www.mercadeo.com/63_saas.htm.

http://www.abcdatos.com/tutoriales/tutorial/z4381.html.

Rincón del Vago. [En línea] [Citado el: 4 de marzo de 2010.] http://pdf.rincondelvago.com/encriptacion-de-datos.html.

Rubio Rincón, Jaime Hernando. escuelaing.edu.cu. [En línea] 2005.

http://profesores.is.escuelaing.edu.co/asignaturas/sypi20071/FOLLETOS%20Y%20MATERIAL%20DE%20ESTUDIO/La%20protecci%F3n%20del%20software.pdf.

Salvador.edu.du. [En línea] http://www.salvador.edu.ar/prjusof.htm .

Siliconkey. [En línea] [Citado el: 2 de marzo de 2010.]

http://www.siliconkey.com.ar/time_protect.htm.

Silver, Mr. Fortunecity. [En línea] [Citado el: 3 de marzo de 2010.]

http://members.fortunecity.com/blackfenix/gguide.html.

Sitepro-sa. [En línea] http://www.sitepro-sa.com.ar/pdf/HARDkey-Descriptivo.pdf.

Taringa. [En línea] [Citado el: 3 de marzo de 2010.]

http://www.taringa.net/posts/downloads/1585542/Compresi%C3%B3n-de-datos.html.

Terra.es. [En línea] [Citado el: 3 de marzo de 2010.]

http://www.terra.es/tecnologia/articulo/html/tec8852.htm.

TortoiseSVN.com. [En línea] http://tortoisesvn.net/.

Trabajo.com. [En línea] http://www.trabajo.com.mx/vision_de_una_empresa.htm.

Transtecnia. [En línea] http://www.transtecnia.cl/contrato_mantencion.htm.

Wapedia. [En línea] [Citado el: 4 de marzo de 2010.]

http://wapedia.mobi/es/Control_de_redundancia_c%C3%ADclica.

Wordpress.com. [En línea] http://ejecucion.wordpress.com/2008/10/09/pasos-para-hacer-estrategia-parte-i/.

GLOSARIO DE TÉRMINOS

1. **Assurance**: Aseguramiento, seguridad, convicción, confianza.
2. **Piratería de software**: Duplicación ilegal del software, es decir, uso del software sin contar con la respectiva licencia.
3. **Content Manager System (CMS)**: Sistema gestor de contenidos. Es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web.
4. **Operations Support System (OSS):** Sistema de soporte de operaciones. Son los sistemas informáticos utilizados por los proveedores de servicios de telecomunicaciones. Describen "los sistemas de red" que trata de la red de telecomunicaciones propia, el apoyo a procesos como el mantenimiento del inventario de red, servicios de aprovisionamiento, la configuración de los componentes de red y la gestión de fallas.
5. **Application Programming Interface (API):**Es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Se utiliza generalmente en las bibliotecas.
6. **Dinamic-Link Library(DLL):**Biblioteca de enlace dinámico. Son archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo.
7. **Criptografía:** Es la ciencia que, mediante el tratamiento de la información, protege a la misma de modificaciones y utilización no autorizada. Utiliza algoritmos matemáticos complejos para la transformación de la información en un extremo y la realización del proceso inverso en el otro extremo.
8. **Encriptación:** Proceso mediante el que se codifica la información, de manera que no pueda leerla ninguna persona no autorizada.
9. **Desencriptación:** Es el proceso de tomar los textos encriptados y la clave criptográfica, y producir un texto claro.
10. **Personal Identification Number (PIN):** Número de Identificación Personal. Es un valor numérico usado para identificarse y poder tener acceso a ciertos sistemas o artefactos, como un teléfono móvil o un cajero automático.
11. **Número de serie:** Es un número alfanumérico único asignado para identificación. Puede constar de un número entero sólo, o contener letras.

12. **Clave:** Es una pieza de información que controla la operación de un algoritmo de criptografía.
13. **Cyclic Redundancy Checking (CRC):** Control de Redundancia Cíclica. Número de control utilizado para comprobar otra serie de valores. Utilizado ampliamente en la tecnología de ordenadores: comprobación del estado de un archivo, chequeo de virus, mecanismos anti copia, protección de datos, generación y chequeo de claves, validación de tarjetas, etc. Su finalidad es comprobar que la operación (transmisión de datos, almacenamiento o recuperación de datos) se han efectuado correctamente.
14. **Banners:** Imagen, gráfico o texto de carácter publicitario, normalmente, de pequeño tamaño que aparece en una página web y que habitualmente enlaza con el sitio web del anunciante.
15. **Dongle:** Hardware de seguridad que se debe conectar al sistema informático antes de que se ejecute una determinada aplicación; previene las copias ilegales de los programas informáticos.
16. **Algoritmo:** Conjunto de instrucciones que permite la resolución de un problema paso a paso. Grupo de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien lo ejecute.
17. **Instalación de un software:** Añadir un programa a un ordenador. Aunque a veces basta con copiarlo al disco duro. Durante el proceso de instalación se crean las carpetas y archivos necesarios para utilizar el determinado programa.
18. **Digital Imaging and Communication in Medicine (DICOM):** Es el estándar reconocido mundialmente para el intercambio de imágenes médicas, pensado para el manejo, almacenamiento, impresión y transmisión de imágenes médicas.
19. **Gestión:** Acción y efecto de administrar.
20. **Procesamiento de imágenes:** Referencia a las aplicaciones de computación en que las imágenes digitalizadas son manipuladas.
21. **Secure Content Management (SCM):** Es un término que designa las soluciones de filtrado de contenido unificado. Se trata de los dispositivos que agrupan funciones de tipo anti-spam, Filtrado Web, Filtrado de Email, Filtrado de URL, etc.
22. **Integer Developer Environment (IDE):** Entorno de Desarrollo Integrado. Se denomina a la suite o conjunto de aplicaciones que conforman un espacio de trabajo pensado para un programador.

23. **Berkeley Software Distribution (BSD):** Distribución de Software Berkeley. Se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.
24. **DEMO:** son programas comerciales que han sido distribuidos de forma gratuita (shareware) con una o más limitaciones respecto a la versión completa.
25. **Acción:** Tarea a realizar para cumplir un objetivo**.**
26. **Objetivos:** Meta o nivel que se debe alcanzar en un período de tiempo determinado.
27. **Técnicas:** Procedimientos o conjunto de reglas, normas o protocolos, que tienen como objetivo obtener un resultado determinado, ya sea en el campo de la ciencia, de la tecnología, del arte, del deporte, de la educación o en cualquier otra actividad.